

**“AÑO DE LA RECUPERACIÓN Y CONSOLIDACIÓN DE LA
ECONOMÍA PERUANA”**

Título del Proyecto: Plataforma de Tutorías Virtuales para
Estudiantes en Riesgo

ODS Vinculado: 4 - Educación de Calidad

ASIGNATURA:

ANÁLISIS Y DISEÑO DE SOFTWARE

NRC: 62152

DOCENTE:

Rosario Delia Osorio Contreras

INTEGRANTES:

- Antezana Alonzo Alexandra Dayana
- Molina Bendezu Leonel
- Tito Llactahuaman Angel Xaviel
- Velasquez Colorado Lionel

Aplicaciones:

Jira: Enlace de jira

Modelio: Enlace de modelio

Canva: Enlace de canva

Github: Enlace

Huancayo, 2025

UNIDAD I – FUNDAMENTOS Y MODELADO INICIAL

CAPÍTULO 1. PRESENTACIÓN DEL PROYECTO

1.1. ODS Vinculado: Educación de Calidad (ODS 4)

La educación es reconocida como un derecho humano fundamental y un elemento indispensable para el desarrollo sostenible, tal como lo establece la Agenda 2030 [1]. Este objetivo busca "garantizar una educación inclusiva, equitativa y de calidad y promover oportunidades de aprendizaje durante toda la vida para todos", con especial énfasis en las poblaciones vulnerables.

1.2. Organización o Institución Beneficiaria

La institución beneficiaria es la **Universidad Continental - Sede Huancayo**. Los usuarios principales son estudiantes de pregrado con bajo rendimiento, docentes que ofrecen tutorías, y coordinadores académicos que supervisan el progreso estudiantil. También puede aplicarse a otras universidades e institutos de la región.

1.3. Problema Identificado

Las desigualdades educativas estructurales persisten en múltiples dimensiones, afectando el acceso, la participación y los resultados de aprendizaje, especialmente en contextos de vulnerabilidad socioeconómica [2]. Estudios recientes demuestran que la falta de personalización en los procesos de enseñanza-aprendizaje contribuye significativamente al bajo rendimiento académico y al abandono escolar temprano [3]. En América Latina, esta problemática se ve agravada por la limitada disponibilidad de sistemas de apoyo educativo personalizado fuera del horario escolar convencional.

1.4. Solución Propuesta

Desarrollamos una plataforma integral de tutorías virtuales que implementa modelos de diferenciación educativa basados en los principios establecidos por Tomlinson [4], donde el contenido, proceso y producto educativo se adaptan a las necesidades específicas de cada estudiante. La plataforma incorpora sistemas de recomendación inteligentes fundamentados en técnicas de filtrado colaborativo y basado en contenido [5], permitiendo emparejamientos óptimos entre tutores y estudiantes.

CAPÍTULO 2. ANÁLISIS DE NECESIDADES Y REQUERIMIENTOS

2.1. Descripción del Problema

El proceso actual de tutorías presenta deficiencias significativas en su organización y seguimiento, lo que genera barreras de acceso al apoyo académico [2]. Estudiantes con bajo rendimiento enfrentan limitaciones para acceder a refuerzo educativo fuera del horario escolar, agravando las desigualdades educativas existentes [3].

Esto genera tres consecuencias críticas:

1. Estudiantes abandonan la búsqueda de ayuda por la complejidad del proceso
2. Tutores potenciales (estudiantes destacados o docentes) no tienen forma de ofrecer sus servicios organizadamente
3. La institución no puede identificar ni intervenir con estudiantes que requieren atención urgente

2.2. Necesidades de los Usuarios

Estudiantes:

- Encontrar tutores calificados rápidamente según la materia
- Agendar sesiones en horarios flexibles sin complicaciones
- Acceder a tutorías desde cualquier ubicación
- Revisar el historial de sus sesiones y materiales compartidos
- Calificar la calidad de las tutorías recibidas

Tutores:

- Publicar su disponibilidad horaria de forma clara
- Gestionar solicitudes de tutoría eficientemente
- Recibir recordatorios de sesiones programadas
- Acceder a materiales o notas previas del estudiante
- Construir reputación mediante valoraciones

Coordinadores Académicos:

- Visualizar reportes de estudiantes en riesgo académico
- Monitorear la frecuencia y efectividad de las tutorías
- Identificar materias con mayor demanda de apoyo
- Exportar datos para análisis institucional

2.3. Requerimientos Funcionales (RF)

Tabla 1. Requerimientos Funcionales del Sistema

Código	Requerimiento	Prioridad
RF01	El sistema debe permitir registro e inicio de sesión con roles (estudiante, tutor, coordinador)	Alta
RF02	El sistema debe permitir búsqueda y agendamiento de sesiones de tutoría por materia y horario	Alta
RF03	El sistema debe integrar videollamadas y enviar notificaciones por correo	Alta
RF04	El sistema debe registrar historial de sesiones y permitir calificar tutores	Media

RF05	El sistema debe generar reportes de estudiantes en riesgo para coordinadores	Alta
------	------------------------------------------------------------------------------	------

2.4. Requerimientos No Funcionales (RNF)

Tabla 2. Requerimientos No Funcionales del Sistema

Código	Requerimiento	Descripción
RNF 01	Usabilidad	Interfaz intuitiva con máximo 3 clics para agendar una sesión
RNF 02	Rendimiento	Tiempo de respuesta menor a 2 segundos para búsquedas
RNF 03	Seguridad	Autenticación mediante JWT y encriptación de contraseñas
RNF 04	Compatibilidad	Responsive y funcional en navegadores principales (Chrome, Firefox, Safari, Edge)

2.5. Requerimientos de Dominio

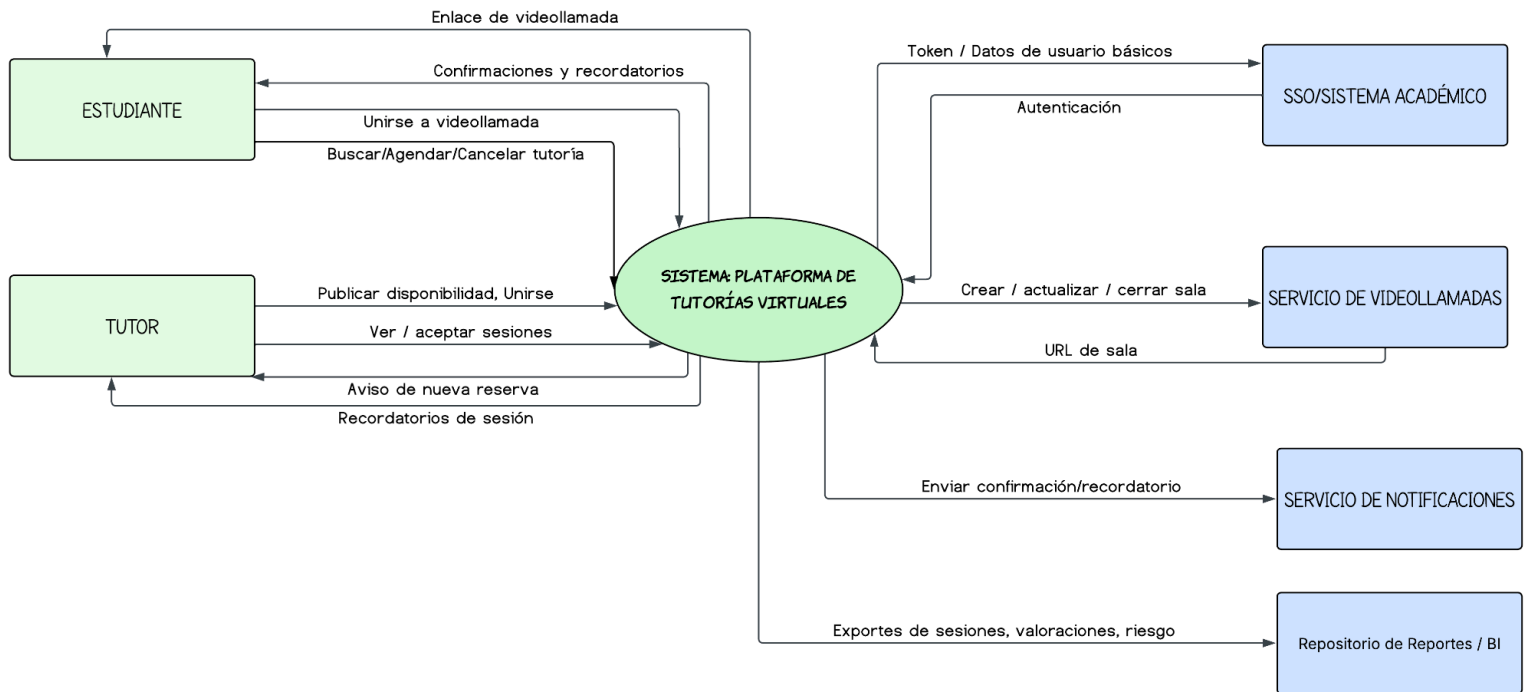
Tabla 3. Requerimientos de Dominio

Código	Requerimiento	Justificación
--------	---------------	---------------

RD0 1	Una sesión de tutoría debe durar mínimo 30 minutos	Estándar académico para sesiones efectivas
RD0 2	Los estudiantes pueden cancelar sesiones con mínimo 2 horas de anticipación	Política institucional de respeto al tiempo del tutor
RD0 3	Los tutores deben tener promedio mínimo de 15 en la materia que enseñan	Requisito de calidad académica
RD0 4	Las sesiones se programan entre 7:00 - 22:00 horas en bloques de 30 minutos	Horario operativo universitario extendido

Capítulo 3. Modelos Iniciales del Sistema

Modelo funcional (diagrama de contexto)



Casos de uso generales:

Modelo de procesos

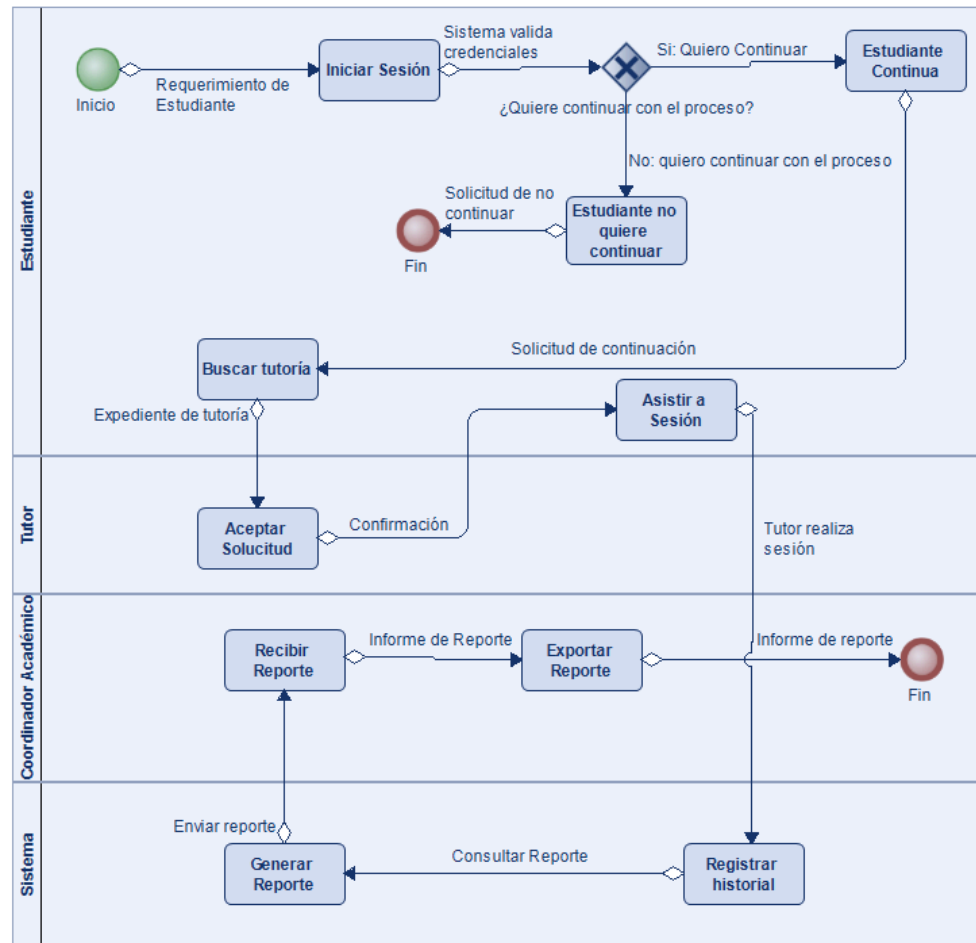


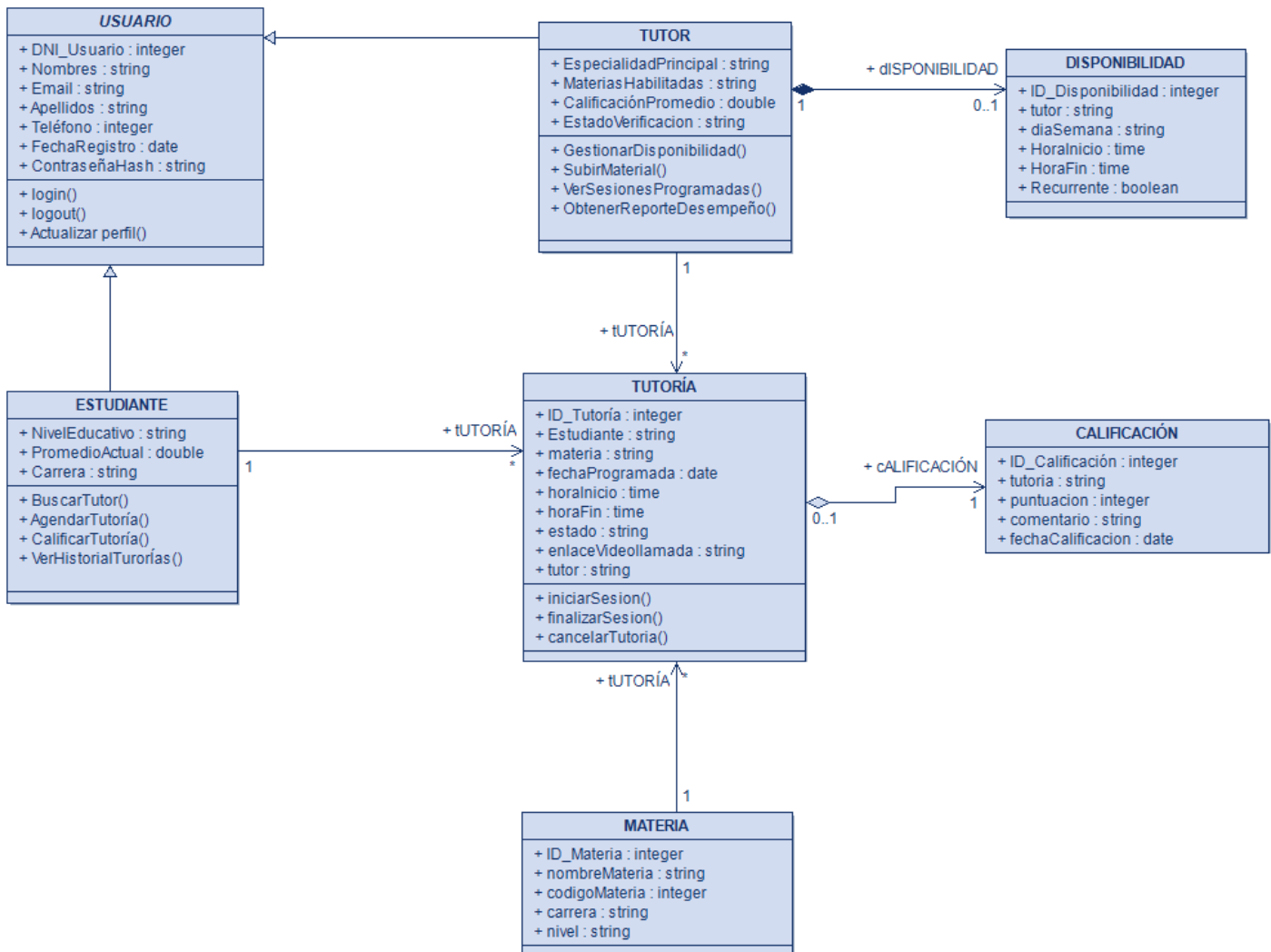
Diagrama de actividad UML

Modelo de datos (Modelo E-R)

Unidad II – Modelos de Diseño y Metodología Ágil (Semanas 5–7)

Capítulo 4. Modelos de Diseño

Modelo estructural (diagrama de clases inicial):



Modelo de interacción (diagrama de secuencia):

Diagrama de secuencia – Agendar tutoría. El Estudiante inicia sesión y busca tutores; la UI envía la solicitud al IncripcionService, que consulta disponibilidad en la base de datos. Al seleccionar tutor y horario, el servicio verifica conflictos: si hay cupo, genera el enlace de videollamada, registra la sesión y envía la confirmación al Estudiante y al Tutor; si no hay cupo, retorna “horario no disponible”.

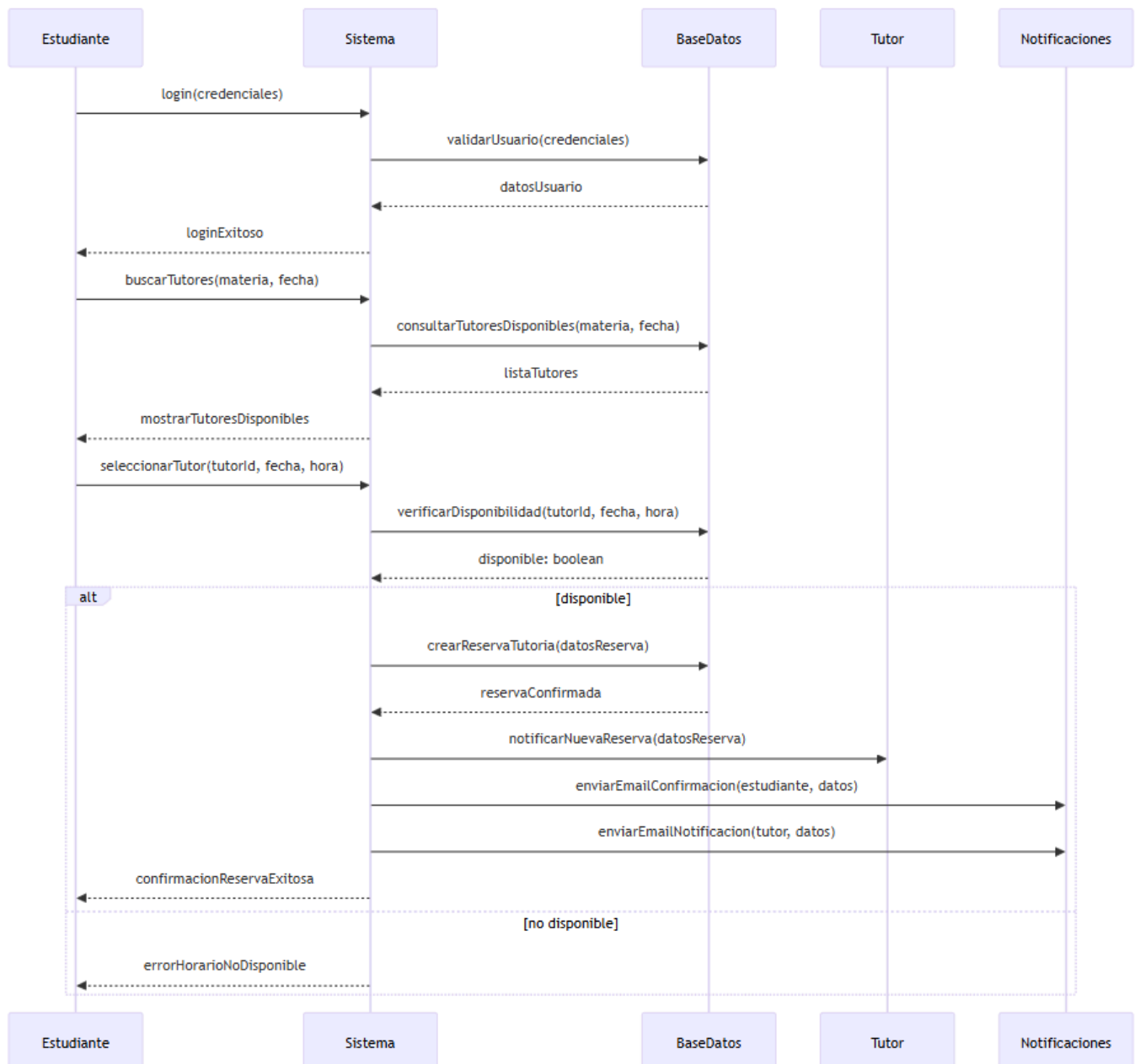
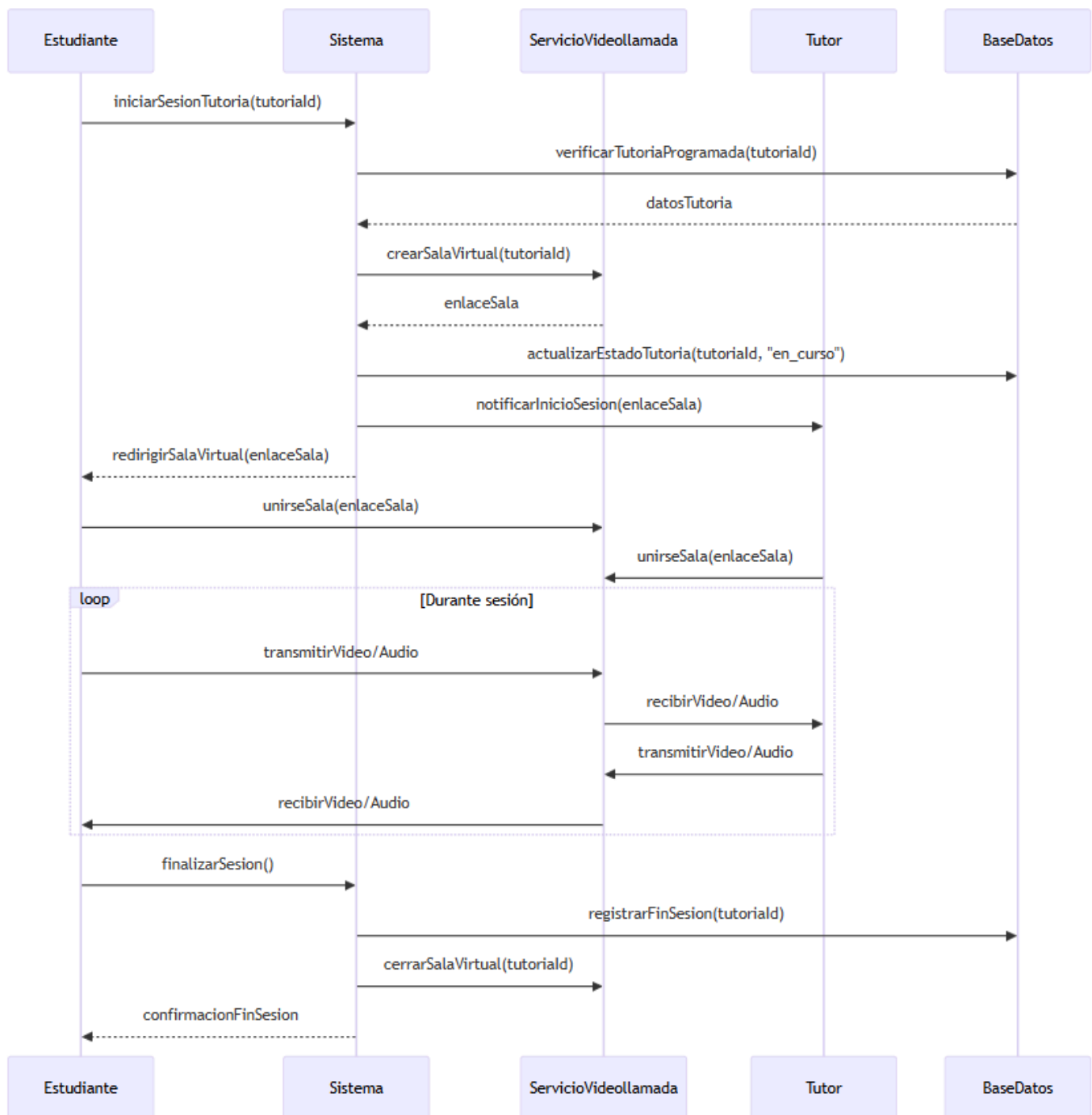


Diagrama de secuencia – Iniciar videollamada de tutoría. El Estudiante solicita iniciar sesión; el Sistema valida la tutoría, crea la sala virtual, redirige con el enlace, ambos participantes se unen, se transmite media (loop) y al finalizar se registra el fin y se cierra la sala.



CAPÍTULO 5. METODOLOGÍA DE TRABAJO (SCRUM)

5.1. Definición de la Metodología Ágil Usada

Para el desarrollo de la Plataforma de Tutorías Virtuales se implementó **SCRUM** como marco de trabajo ágil. SCRUM permite entregas incrementales del producto mediante ciclos cortos llamados sprints, facilitando la adaptación a cambios y la retroalimentación continua.

Roles del equipo:

- **Product Owner:** Coordinador académico que define prioridades del backlog
- **Scrum Master:** Rosario Osorio Contreras (facilita el proceso y elimina impedimentos)
- **Development Team:** Antezana Alexandra, Molina Leonel, Tito Angel, Velasquez Lionel

Eventos SCRUM implementados:

- Sprint Planning: Planificación al inicio de cada sprint (2 semanas)
- Daily Stand-up: Reuniones diarias de 15 minutos (virtuales)
- Sprint Review: Demostración del incremento al finalizar el sprint
- Sprint Retrospective: Análisis de mejoras del proceso

5.2. Backlog del Producto (Épicas e Historias de Usuario)

Épicas del Producto

EP01 - Gestión de Usuarios Permite el registro, autenticación y administración de perfiles de estudiantes, tutores y coordinadores.

EP02 - Sistema de Tutorías Facilita la búsqueda, agendamiento y ejecución de sesiones de tutoría virtual.

EP03 - Seguimiento y Reportes Proporciona herramientas de monitoreo y análisis para coordinadores académicos.

Historias de Usuario Priorizadas

ID	Historia de Usuario	Épica	Prioridad	Puntos
HU01	Como estudiante quiero registrarme en el sistema para acceder a las tutorías	EP01	Alta	5
HU02	Como estudiante quiero buscar tutores por materia para encontrar ayuda específica	EP02	Alta	8
HU03	Como estudiante quiero agendar una sesión para recibir tutoría en horario disponible	EP02	Alta	13
HU04	Como tutor quiero publicar mi disponibilidad para que estudiantes me encuentren	EP01	Alta	8
HU05	Como estudiante quiero recibir notificaciones por correo para recordar mis sesiones	EP02	Media	5
HU06	Como tutor quiero acceder a videollamada integrada para realizar la tutoría virtual	EP02	Alta	13
HU07	Como estudiante quiero calificar al tutor para ayudar a otros estudiantes	EP02	Media	5
HU08	Como coordinador quiero ver reportes de estudiantes en riesgo para intervenir oportunamente	EP03	Alta	13

Criterios de Aceptación (Ejemplo - HU03):

- El estudiante puede seleccionar fecha y hora del calendario del tutor
- El sistema valida que no haya conflictos de horario
- Se envía confirmación automática por correo a ambas partes
- La sesión queda registrada en el historial del estudiante

5.3. Planificación de Sprints

Sprint 1 (Semanas 1-2)

Objetivo: Implementar funcionalidades básicas de gestión de usuarios y búsqueda de tutores

Historias incluidas:

- HU01: Registro de usuarios (5 puntos)
- HU02: Búsqueda de tutores (8 puntos)
- HU04: Publicación de disponibilidad (8 puntos)

Capacidad del equipo: 21 puntos de historia

Entregables:

- Módulo de autenticación funcional
- Base de datos con tablas de usuarios y disponibilidad
- Interfaz de búsqueda de tutores con filtros

Definición de Done:

- Código revisado y aprobado por el equipo
- Pruebas unitarias ejecutadas exitosamente
- Funcionalidad desplegada en entorno de desarrollo
- Documentación técnica actualizada

Sprint 2 (Semanas 3-4)

Objetivo: Desarrollar sistema de agendamiento y videollamadas

Historias incluidas:

- HU03: Agendamiento de sesiones (13 puntos)
- HU05: Notificaciones por correo (5 puntos)
- HU06: Integración de videollamadas (13 puntos)

Capacidad del equipo: 31 puntos de historia

Entregables:

- Calendario interactivo de agendamiento
- Sistema de notificaciones automáticas

- Integración con API de Google Meet
- Módulo de gestión de sesiones

Riesgos identificados:

- Complejidad de integración con API externa de videollamadas
- Posibles problemas de sincronización de horarios

Mitigación:

- Investigación previa de documentación de APIs
- Implementación de sistema de logs para debugging

5.4. Herramientas Utilizadas

Gestión de Proyecto:

- **Jira:** Gestión del backlog, sprints y seguimiento de historias de usuario
- **Confluence:** Documentación técnica y actas de reuniones

Diseño y Modelado:

- **Draw.io:** Creación de diagramas UML (casos de uso, secuencia, actividad)
- **Dbdiagram.io:** Diseño del modelo entidad-relación de la base de datos
- **Figma:** Prototipado de interfaces de usuario

Desarrollo:

- **Visual Studio Code:** IDE principal para desarrollo
- **GitHub:** Control de versiones y colaboración en código
- **Postman:** Pruebas de APIs REST

Comunicación:

- **Discord:** Reuniones diarias y comunicación del equipo
 - **Google Meet:** Sprint reviews y retrospectivas
 - **WhatsApp:** Coordinación rápida y notificaciones urgentes
-

CONCLUSIONES Y RECOMENDACIONES

Conclusiones del Equipo

1. La Plataforma de Tutorías Virtuales representa una solución efectiva para democratizar el acceso al apoyo académico, eliminando barreras geográficas y temporales que afectan a estudiantes en riesgo.
2. La aplicación de la metodología SCRUM permitió al equipo trabajar de forma organizada con entregas incrementales, facilitando la adaptación a cambios y la detección temprana de problemas técnicos.
3. El análisis detallado de requerimientos fue fundamental para identificar las necesidades reales de los tres tipos de usuarios (estudiantes, tutores y coordinadores), evitando funcionalidades innecesarias.
4. La integración de tecnologías modernas como React.js, Node.js y APIs de videollamadas garantiza que la plataforma sea escalable y mantenible a largo plazo.
5. El proyecto contribuye directamente al ODS 4 al mejorar la calidad educativa mediante herramientas tecnológicas accesibles que reducen la deserción universitaria.

Lecciones Aprendidas

1. **Comunicación constante:** Las reuniones diarias fueron clave para mantener al equipo sincronizado y resolver impedimentos rápidamente.
2. **Priorización efectiva:** Definir correctamente las historias de usuario de alta prioridad permitió entregar valor desde los primeros sprints.
3. **Investigación previa:** Dedicar tiempo a investigar las APIs de videollamadas antes del Sprint 2 evitó bloqueos técnicos durante el desarrollo.
4. **Documentación continua:** Mantener actualizada la documentación en Confluence facilitó la incorporación de nuevos miembros y la consulta de decisiones previas.
5. **Flexibilidad:** Ser capaces de adaptar el plan cuando surgieron imprevistos fue esencial para cumplir los objetivos del proyecto.

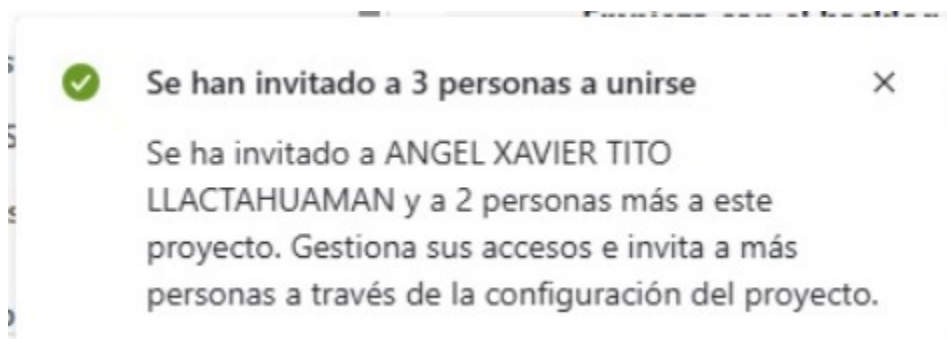
Recomendaciones para Futuras Mejoras del Sistema

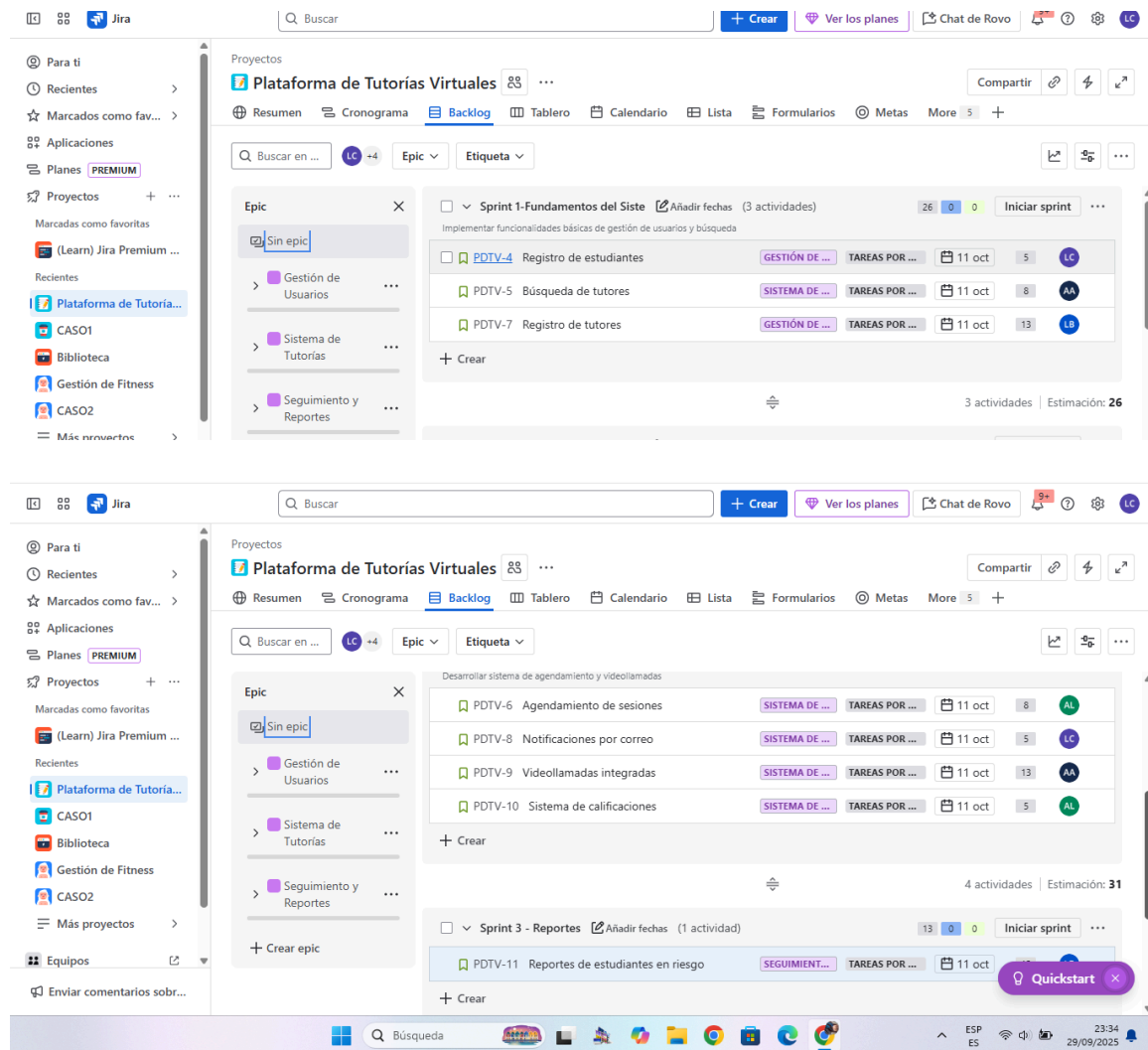
1. **Inteligencia Artificial:** Implementar un sistema de recomendación que sugiera tutores basándose en el historial académico y preferencias del estudiante.
2. **Gamificación:** Agregar insignias y puntos para tutores destacados, incentivando la participación y calidad del servicio.
3. **Aplicación móvil nativa:** Desarrollar versiones para iOS y Android que mejoren la experiencia del usuario en dispositivos móviles.
4. **Sistema de pagos:** Incorporar pasarela de pagos para tutorías privadas fuera del programa institucional, generando un modelo de negocio sostenible.
5. **Analítica avanzada:** Implementar dashboards con métricas predictivas que identifiquen estudiantes en riesgo antes de que reprueben asignaturas.
6. **Integración con LMS:** Conectar la plataforma con sistemas de gestión académica (Canvas, Moodle) para sincronizar automáticamente las notas y detectar necesidades de tutoría.
7. **Grabación de sesiones:** Permitir grabar tutorías (con consentimiento) para que los estudiantes puedan repasar el contenido posteriormente.
8. **Chat en tiempo real:** Agregar mensajería instantánea para consultas rápidas que no requieran una sesión completa de videollamada.

Referencias bibliográficas

Anexos

Evidencias gráficas (capturas de Jira, capturas de GITHUB y commits, evidencias de trabajo en equipo).





Referencias bibliográficas (ISO 690 numérico).

[1] SACHS, J. D. La era del desarrollo sostenible. Barcelona: Deusto, 2015. 576 p. ISBN 9788423419939.

[2] UNESCO. Informe de seguimiento de la educación en el mundo 2020: Inclusión y educación: todos sin excepción. París: UNESCO, 2020. 435 p. ISBN 9789233000885.

[3] HARGREAVES, A. y FULLAN, M. Capital profesional: Transformar la enseñanza en cada escuela. Madrid: Morata, 2014. 256 p. ISBN 9788471128049.

[4] TOMLINSON, C. A. El aula diversificada: Dar respuesta a las necesidades de todos los estudiantes. Barcelona: Octaedro, 2015. 216 p. ISBN 9788499216960.