



"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

Huancayo, 2025

Aplicaciones

Jira: <https://continental-team-tebflnru.atlassian.net/jira/software/projects/PDTV/boards/100/backlog?epics=visible&selectedIssue=PDTV-11&atlOrigin=eyJpIjoiYjUxZDg0NDhmNTE4NDA4MGE2ODhjYTVIYTQ3NTc2ZDkiLCJwljoiaiJ9>

Canva: https://www.canva.com/design/DAG6O3zxOGw/gBYxVuD5S0FGNfZ3PIBE2A/edit?utm_content=DAG6O3zxOGw&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Git:

https://github.com/AlexandraAntezana-/Plataforma-de-Tutor-as-Virtuales-para-Estudiantes-en-Riesgo_ODS4

Figma:

<https://www.figma.com/design/3og0Vvb2oHJ16DIzoot6/Untitled?node-id=0-1&t=q13JwTECCZ1TNgKe-1>

ÍNDICE

UNIDAD I – FUNDAMENTOS Y MODELADO INICIAL

CAPÍTULO 1. PRESENTACIÓN DEL PROYECTO

- 1.1. ODS Vinculado: Educación de Calidad (ODS 4)
- 1.2. Organización o Institución Beneficiaria
- 1.3. Problema Identificado
- 1.4. Solución Propuesta

CAPÍTULO 2. ANÁLISIS DE NECESIDADES Y REQUERIMIENTOS

- 2.1. Descripción del Problema
- 2.2. Necesidades de los Usuarios
- 2.3. Requerimientos Funcionales (RF)
- 2.4. Requerimientos No Funcionales (RNF)
- 2.5. Requerimientos de Dominio

CAPÍTULO 3. MODELOS INICIALES DEL SISTEMA

- Modelo funcional (diagrama de contexto)

- Casos de uso generales
- Modelo de procesos
- Modelo de datos (Modelo E-R)

UNIDAD II – MODELOS DE DISEÑO Y METODOLOGÍA ÁGIL

CAPÍTULO 4. MODELOS DE DISEÑO

- Modelo estructural (diagrama de clases inicial)
- Modelo de interacción (diagrama de secuencia)

CAPÍTULO 5. METODOLOGÍA DE TRABAJO (SCRUM)

- 5.1. Definición de la Metodología Ágil Usada
- 5.2. Backlog del Producto (Épicas e Historias de Usuario)
- 5.3. Planificación de Sprints
- 5.4. Herramientas Utilizadas

CONCLUSIONES Y RECOMENDACIONES

- Conclusiones del Equipo
- Lecciones Aprendidas
- Recomendaciones para Futuras Mejoras del Sistema

REFERENCIAS BIBLIOGRÁFICAS

ANEXOS

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. 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

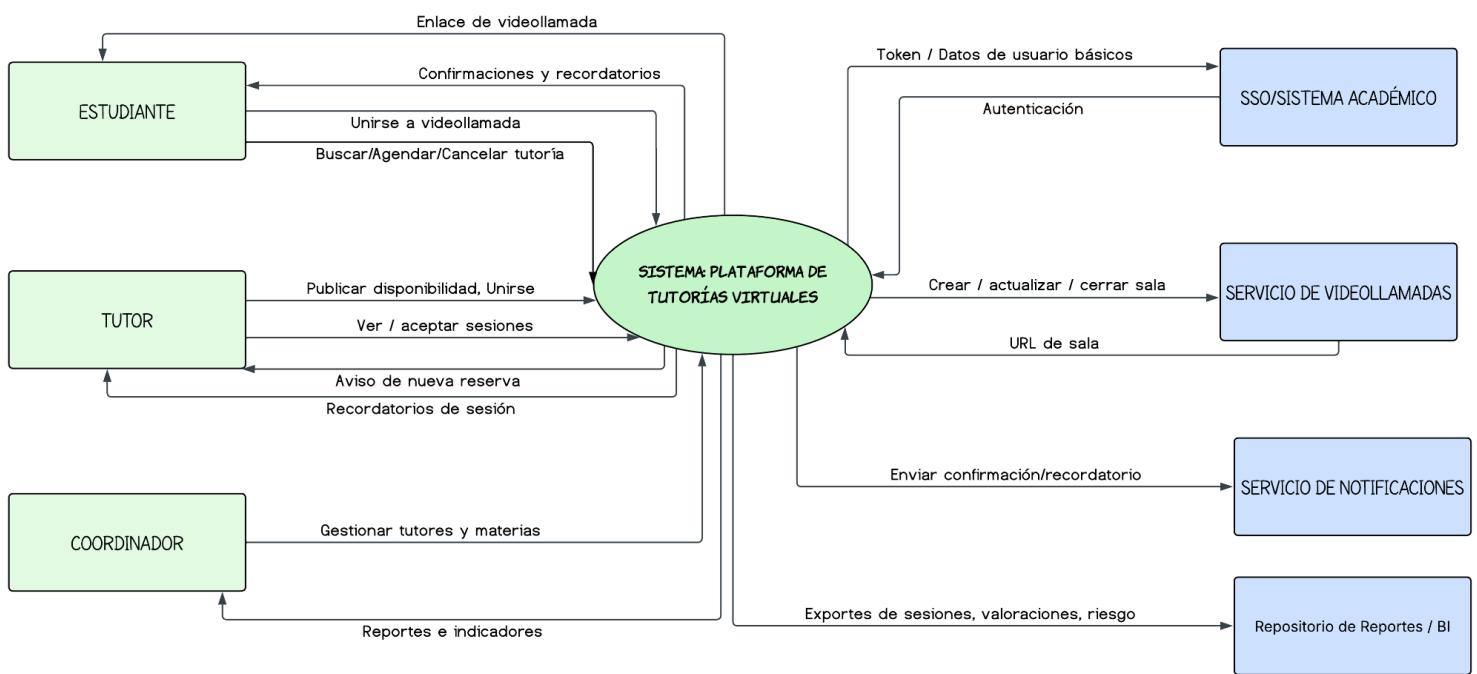
Tabla 3. Requerimientos de Dominio

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

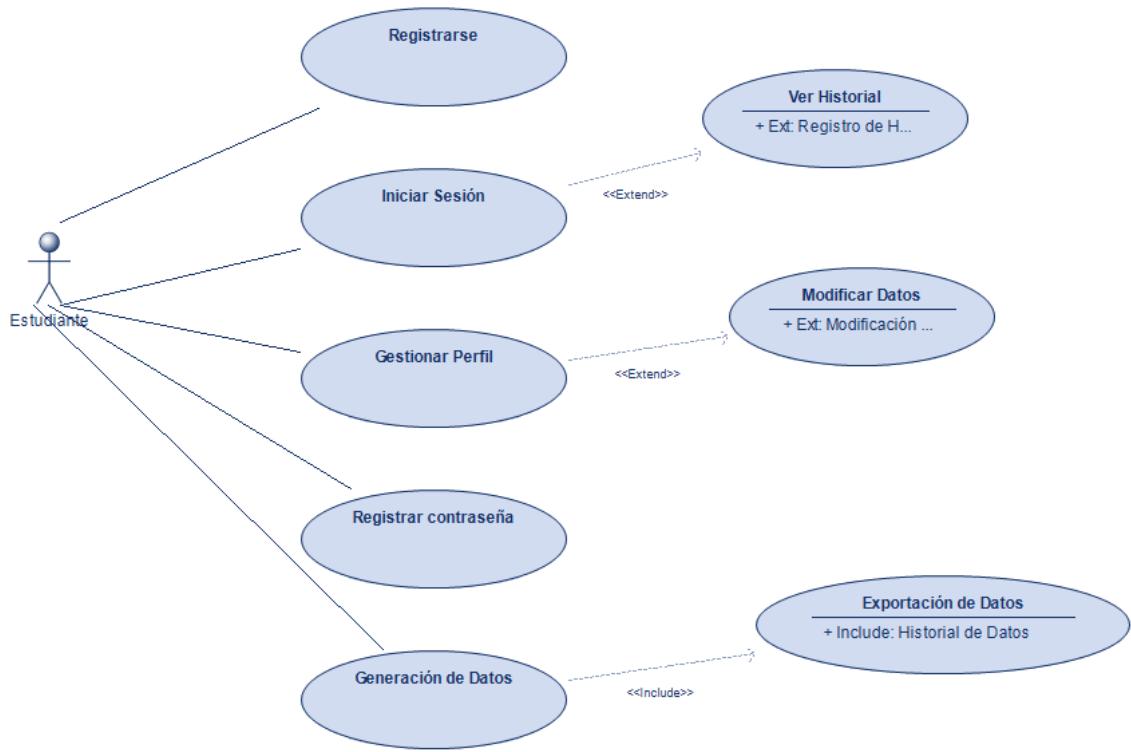
RD0 3	Los tutores deben tener promedio mínimo de 15 en la materia que enseñan	Requisito de calidad académica
RD0 4	Las sesiones se programan entre 7:00 - 22:00 horas en bloques de 30 minutos	Horario operativo universitario extendido

Capítulo 3. Modelos Iniciales del Sistema

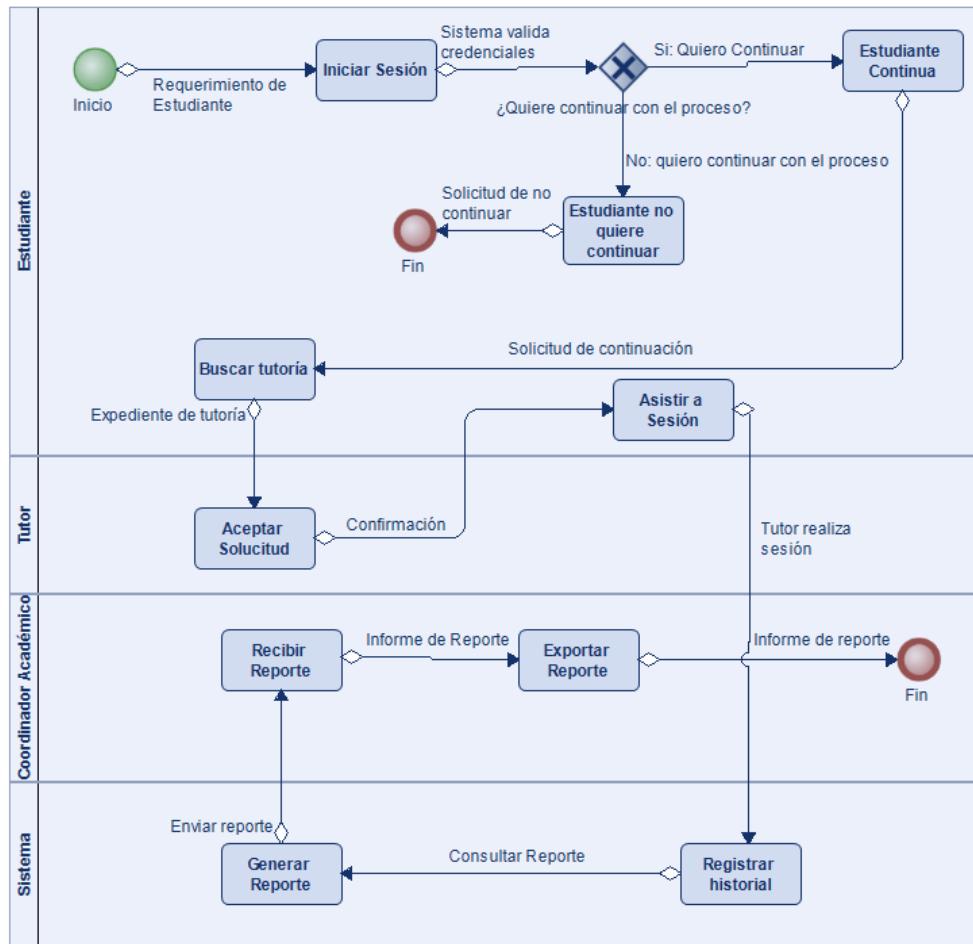
Modelo funcional (diagrama de contexto)



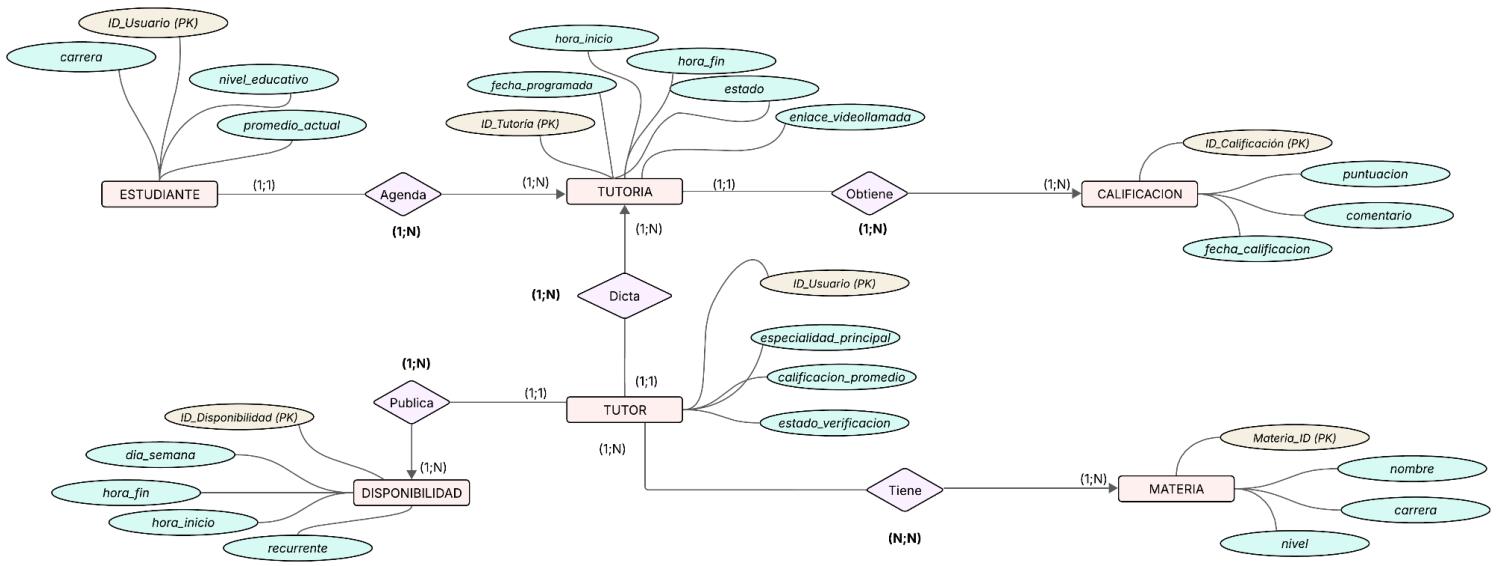
Casos de uso generales:



Modelo de procesos



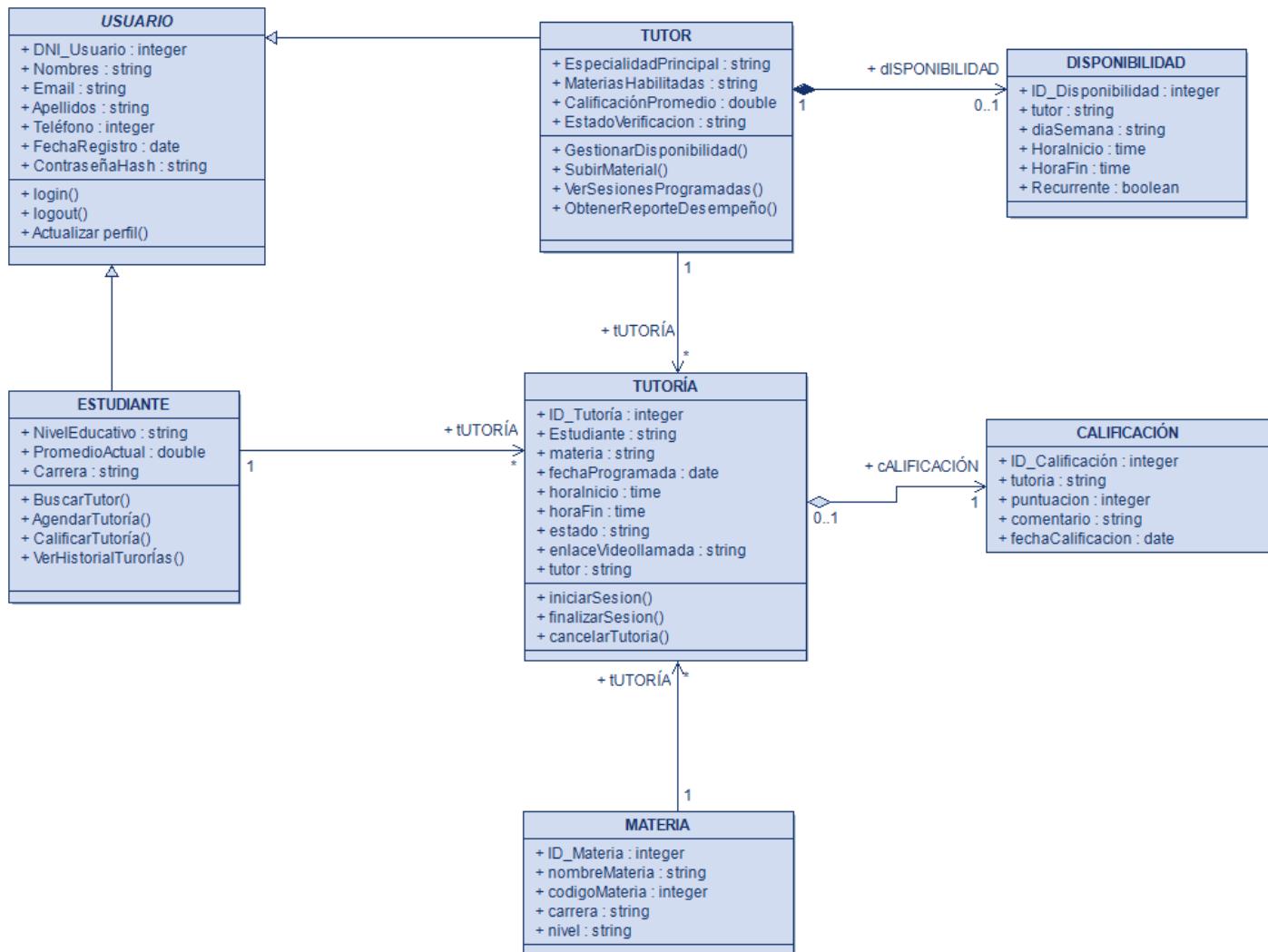
Modelo de datos (Modelo E-R)



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 InscriptioService, 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”.

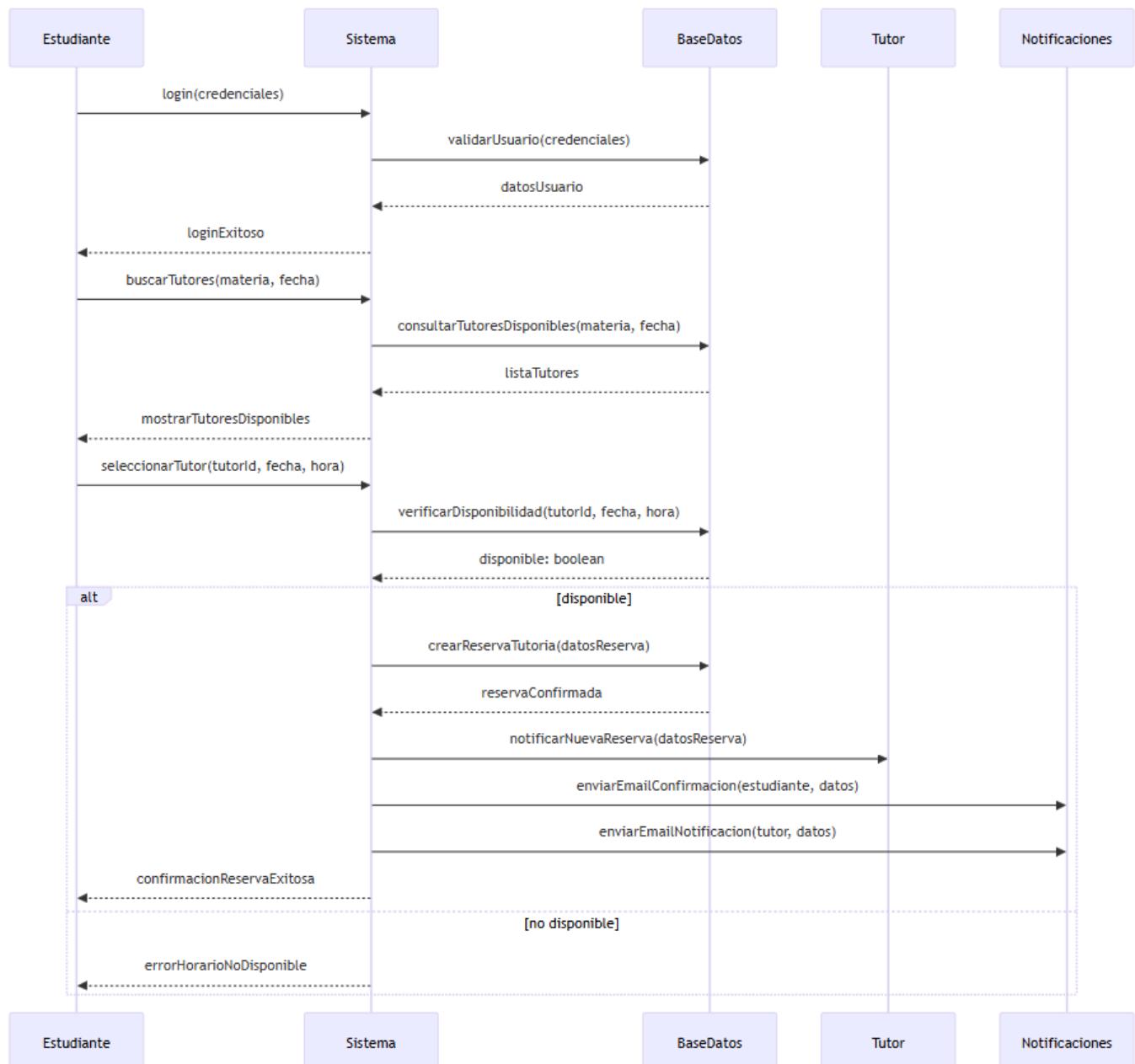
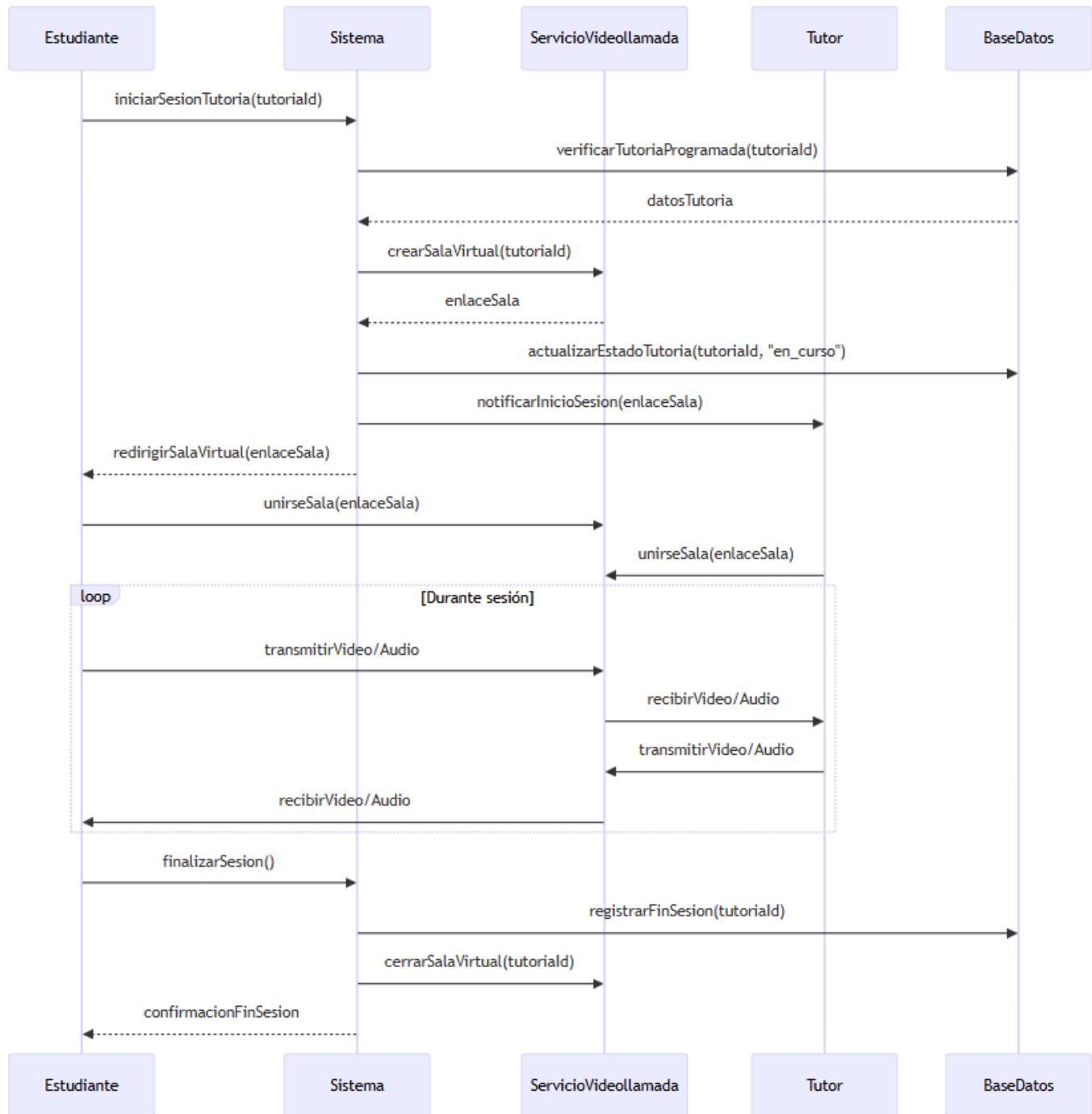


Diagrama de secuencia – Iniciar videollamada de tutoría. El Estudiante solicita iniciar sesión; el Sistema valida la tutoría, crea la sala virtual, redirige con el enlace, ambos participantes se unen, se transmite media (loop) y al finalizar se registra el fin y se cierra la sala.



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
HU 01	Como estudiante quiero registrarme en el sistema para acceder a las tutorías	EP0 1	Alta	5
HU 02	Como estudiante quiero buscar tutores por materia para encontrar ayuda específica	EP0 2	Alta	8
HU 03	Como estudiante quiero agendar una sesión para recibir tutoría en horario disponible	EP0 2	Alta	13
HU 04	Como tutor quiero publicar mi disponibilidad para que estudiantes me encuentren	EP0 1	Alta	8
HU 05	Como estudiante quiero recibir notificaciones por correo para recordar mis sesiones	EP0 2	Media	5
HU 06	Como tutor quiero acceder a videollamada integrada para realizar la tutoría virtual	EP0 2	Alta	13
HU 07	Como estudiante quiero calificar al tutor para ayudar a otros estudiantes	EP0 2	Media	5
HU 08	Como coordinador quiero ver reportes de estudiantes en riesgo para intervenir oportunamente	EP0 3	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

Unidad III – Diseño de Software

Capítulo 6. Diseño de Arquitectura y Patrones

6.1 Estrategia de diseño del software

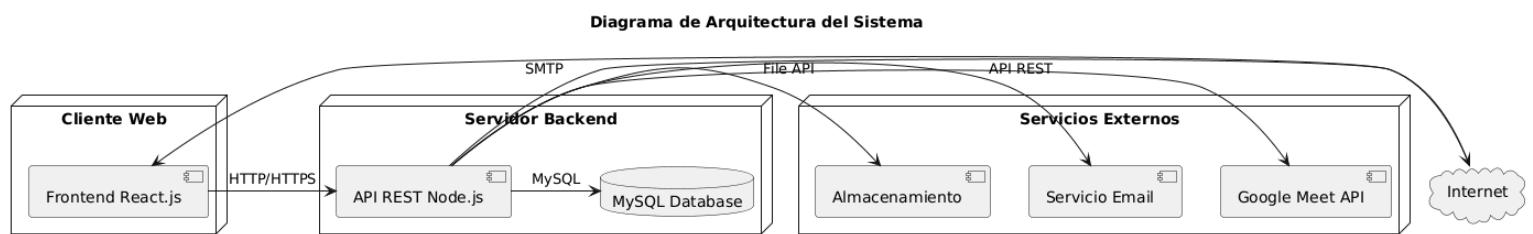
La estrategia de diseño adoptada para la Plataforma de Tutorías Virtuales sigue un enfoque basado en componentes con separación clara de responsabilidades. Se implementa una arquitectura en capas que facilita el mantenimiento, escalabilidad y testabilidad del sistema.

6.2 Tipo de arquitectura del sistema

Arquitectura de Referencia: MVC (Modelo-Vista-Controlador) en el frontend y Arquitectura en Capas (Layered Architecture) en el backend.

Justificación:

- Separación de preocupaciones: Cada capa tiene responsabilidades específicas
- Escalabilidad: Permite escalar componentes de forma independiente
- Mantenibilidad: Facilita la implementación de cambios y correcciones
- Testabilidad: Cada capa puede ser probada de forma aislada



6.3 Patrones de diseño aplicados

1. Patrón Repository

- Aplicación: Acceso a datos en la capa de persistencia
- Beneficio: Abstraction de la lógica de acceso a datos
- Implementación: Clases Repository para cada entidad principal (Usuario, Tutoría, Sesión)

2. Patrón Factory

- Aplicación: Creación de objetos complejos (notificaciones, sesiones de videollamada)
- Beneficio: Centraliza la lógica de creación
- Implementación: NotificationFactory, SessionFactory

3. Patrón Strategy

- Aplicación: Algoritmos de recomendación de tutores
- Beneficio: Intercambiable entre diferentes estrategias
- Implementación: TutorMatchingStrategy interface con implementaciones concretas

4. Patrón Observer

- Aplicación: Sistema de notificaciones
- Beneficio: Desacoplamiento entre emisores y receptores de eventos
- Implementación: Eventos de sistema (nueva sesión, recordatorio, calificación)

5. Patrón Singleton

- Aplicación: Conexión a base de datos, configuración del sistema
- Beneficio: Una única instancia global
- Implementación: DatabaseConnection, AppConfig

6.4 Diseño estructural

Diagrama de Componentes:



Capítulo 7. Diseño Detallado de la Base de Datos

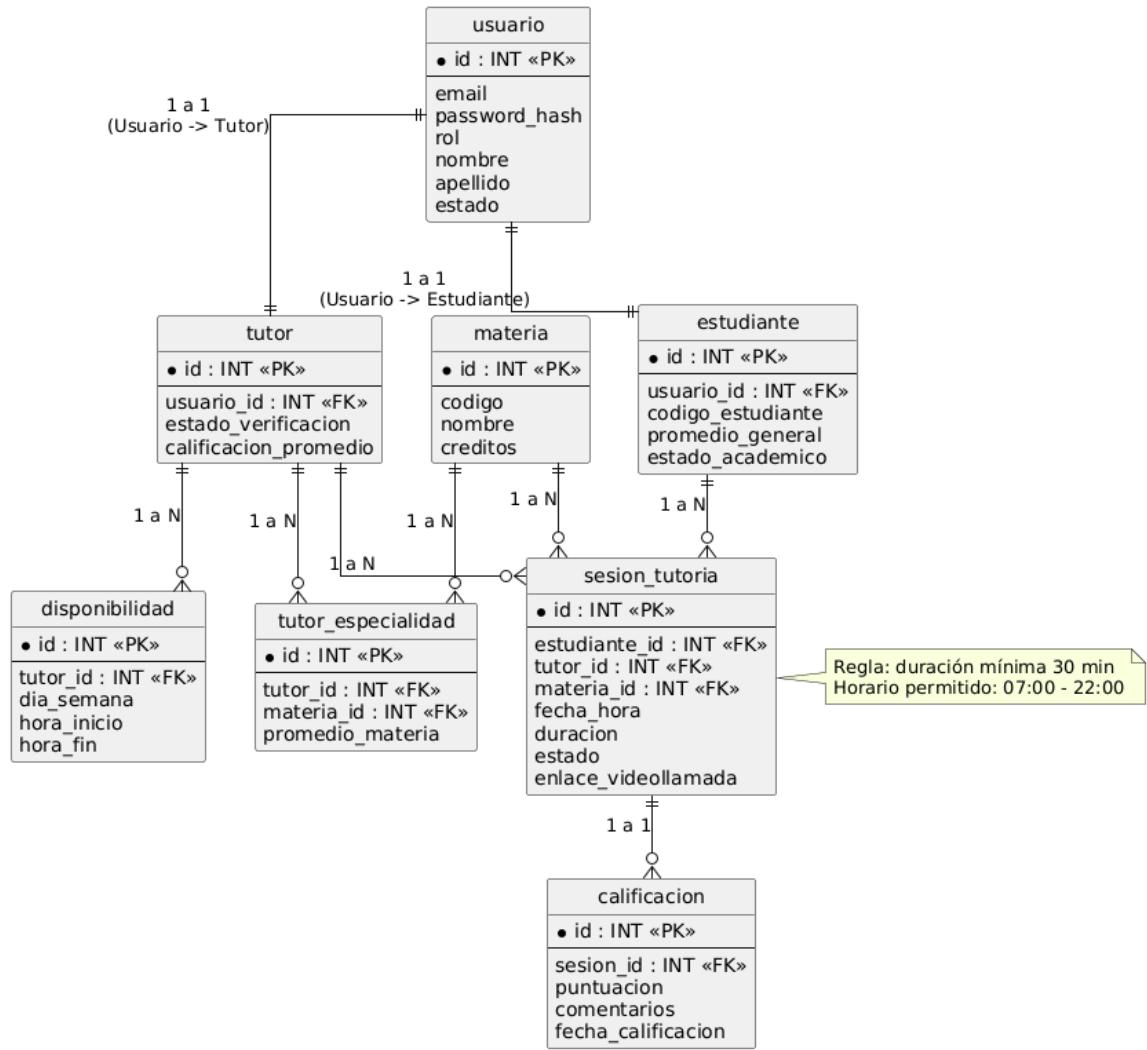
7.1 Modelo Lógico y Físico

El diseño de la base de datos se elaboró en base a los requerimientos funcionales del sistema de tutorías. Se identificaron las entidades principales, sus atributos y las relaciones entre ellas, generando un modelo lógico completamente normalizado.

Las entidades clave del sistema son:

- **Usuario**
- **Estudiante**
- **Tutor**
- **Materia**
- **TutorEspecialidad**
- **Disponibilidad**
- **SesionTutoria**
- **Calificacion**

7.1.1 Modelo Lógico (3FN)



El modelo lógico describe todas las entidades con sus claves primarias, claves foráneas y cardinalidades. Las relaciones se estructuran en:

- Un usuario puede ser estudiante, tutor o coordinador.
- Un tutor puede enseñar varias materias → relación N a M mediante **TutorEspecialidad**.
- Un estudiante puede agendar múltiples sesiones.

- Un tutor tiene horarios registrados en **Disponibilidad**.
 - Una sesión opcionalmente tiene una calificación asociada.
-

7.1.2 Proceso de Normalización hasta 3FN

1FN – Eliminación de atributos multivalorados

- El atributo *especialidades* del tutor fue separado en la tabla **TutorEspecialidad**.
- Todas las tablas tienen valores atómicos.

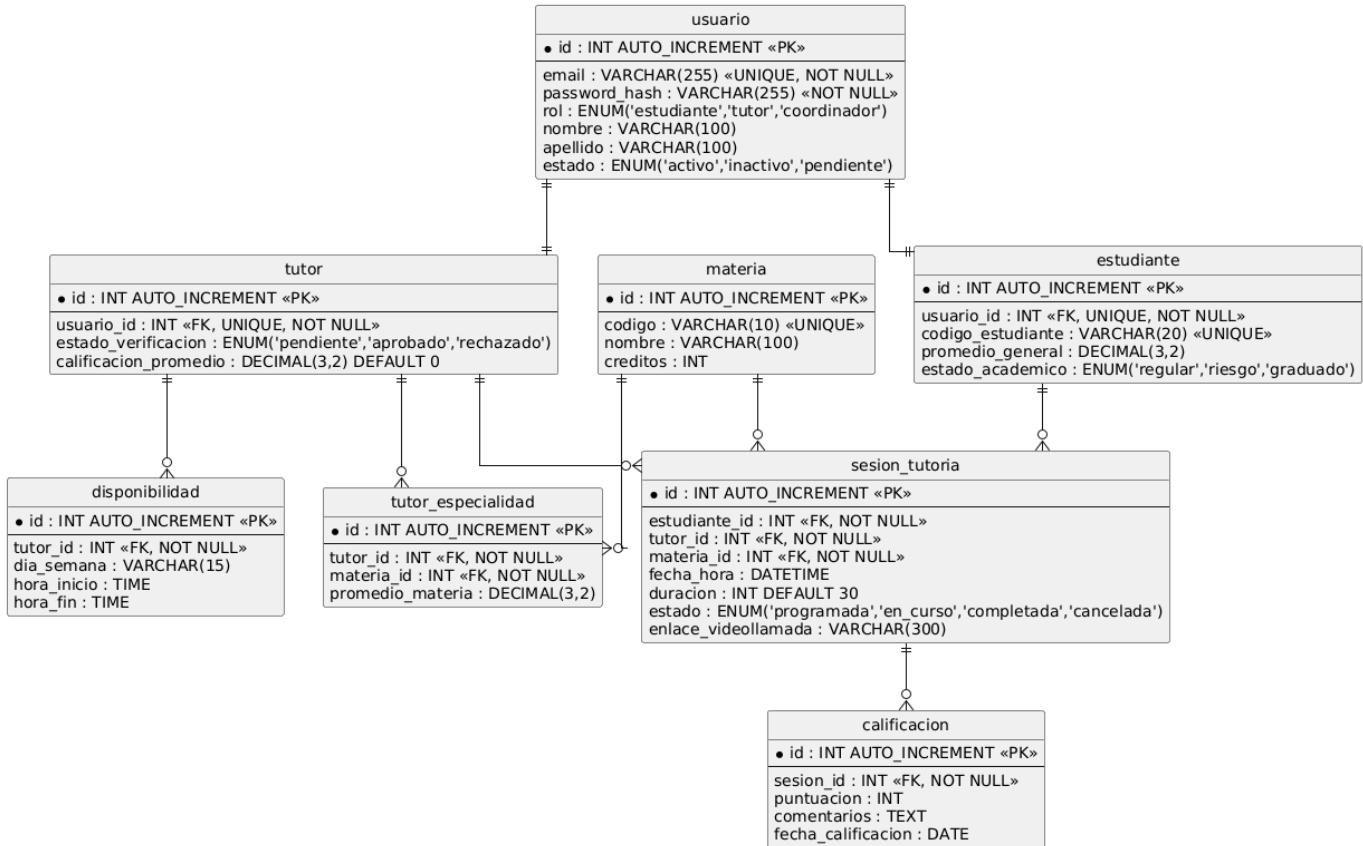
2FN – Eliminación de dependencias parciales

- Los datos de materias se separan en entidad propia.
- La disponibilidad del tutor se maneja en una tabla independiente.

3FN – Eliminación de dependencias transitivas

- No se almacenan datos calculables (ej. promedio del tutor) en las entidades principales.
 - Los datos derivados se administran mediante triggers.
-

7.1.3 Modelo Físico



Se definieron:

- Tipos de datos (VARCHAR, DECIMAL, ENUM, DATETIME)
- Claves primarias y foráneas
- Restricciones de integridad (UNIQUE, NOT NULL)
- Índices en campos de búsqueda frecuente

7.2 Script SQL Resumido de Creación de Tablas

-- Creación de la Base de Datos

```
CREATE TABLE usuario (
    id INT AUTO_INCREMENT PRIMARY KEY,
    email VARCHAR(255) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    rol ENUM('estudiante','tutor','coordinador') NOT NULL,
    nombre VARCHAR(100),
    apellido VARCHAR(100),
    estado ENUM('activo','inactivo','pendiente') DEFAULT 'pendiente'
);
```

```
CREATE TABLE estudiante (
    id INT AUTO_INCREMENT PRIMARY KEY,
    usuario_id INT UNIQUE NOT NULL,
    codigo_estudiante VARCHAR(20) UNIQUE,
    promedio_general DECIMAL(3,2),
    estado_academico ENUM('regular','riesgo','graduado'),
    FOREIGN KEY (usuario_id) REFERENCES usuario(id)
);
```

```
CREATE TABLE tutor (
    id INT AUTO_INCREMENT PRIMARY KEY,
    usuario_id INT UNIQUE NOT NULL,
    estado_verificacion ENUM('pendiente','aprobado','rechazado'),
    calificacion_promedio DECIMAL(3,2) DEFAULT 0,
    FOREIGN KEY (usuario_id) REFERENCES usuario(id)
);
```

```
CREATE TABLE materia (
    id INT AUTO_INCREMENT PRIMARY KEY,
    codigo VARCHAR(10) UNIQUE,
    nombre VARCHAR(100),
    creditos INT
);
```

```
CREATE TABLE tutor_especialidad (
    id INT AUTO_INCREMENT PRIMARY KEY,
    tutor_id INT,
    materia_id INT,
    promedio_materia DECIMAL(3,2),
    FOREIGN KEY (tutor_id) REFERENCES tutor(id),
    FOREIGN KEY (materia_id) REFERENCES materia(id)
);
```

```
CREATE TABLE disponibilidad (
    id INT AUTO_INCREMENT PRIMARY KEY,
    tutor_id INT,
    dia_semana VARCHAR(15),
    hora_inicio TIME,
    hora_fin TIME,
    FOREIGN KEY (tutor_id) REFERENCES tutor(id)
);
```

```
CREATE TABLE sesion_tutoria (
    id INT AUTO_INCREMENT PRIMARY KEY,
    estudiante_id INT,
    tutor_id INT,
    materia_id INT,
    fecha_hora DATETIME,
    duracion INT DEFAULT 30,
    estado ENUM('programada','en_curso','completada','cancelada'),
    enlace_videollamada VARCHAR(300),
    FOREIGN KEY (estudiante_id) REFERENCES estudiante(id),
    FOREIGN KEY (tutor_id) REFERENCES tutor(id),
    FOREIGN KEY (materia_id) REFERENCES materia(id)
);
```

```
CREATE TABLE calificacion (
    id INT AUTO_INCREMENT PRIMARY KEY,
    sesion_id INT,
    puntuacion INT,
    comentarios TEXT,
    fecha_calificacion DATE,
    FOREIGN KEY (sesion_id) REFERENCES sesion_tutoria(id)
);
```

7.3 Procedimientos Almacenados

7.3.1 SP para Agendar Sesión

```

DELIMITER $$

CREATE PROCEDURE sp_agendar_sesion(
    IN p_estudiante INT,
    IN p_tutor INT,
    IN p_materia INT,
    IN p_fecha DATETIME
)
BEGIN
    IF EXISTS (SELECT 1 FROM sesion_tutoria
        WHERE tutor_id = p_tutor
        AND fecha_hora = p_fecha
        AND estado = 'programada') THEN
        SELECT 'conflicto' AS estado;
    ELSE
        INSERT INTO
            sesion_tutoria(estudiante_id,tutor_id,materia_id,fecha_hora,estado)
        VALUES(p_estudiante,p_tutor,p_materia,p_fecha,'programada');
        SELECT 'ok' AS estado;
    END IF;
END$$

DELIMITER ;

```

7.3.2 SP para Reporte Básico de Estudiantes en Riesgo

```

DELIMITER $$

CREATE PROCEDURE sp_estudiantes_riesgo()

```

```
BEGIN  
  
    SELECT e.id, u.nombre, u.apellido, e.promedio_general  
    FROM estudiante e  
  
    JOIN usuario u ON u.id = e.usuario_id  
  
    WHERE e.estado_academico = 'riesgo';  
  
END$$  
  
DELIMITER ;
```

7.4 Vistas de Consulta

7.4.1 Vista: Tutores disponibles

```
CREATE VIEW vw_tutores_disponibles AS  
  
SELECT t.id AS tutor_id, u.nombre, u.apellido, m.nombre AS materia  
FROM tutor t  
  
JOIN usuario u ON t.usuario_id = u.id  
  
JOIN tutor_especialidad te ON te.tutor_id = t.id  
  
JOIN materia m ON m.id = te.materia_id  
  
WHERE t.estado_verificacion = 'aprobado';
```

7.4.2 Vista: Sesiones programadas

```
CREATE VIEW vw_sesiones_programadas AS  
  
SELECT s.id, e.codigo_estudiante, u.nombre AS tutor, m.nombre AS materia,  
s.fecha_hora  
FROM sesion_tutoria s  
  
JOIN estudiante e ON e.id = s.estudiante_id  
  
JOIN tutor t ON t.id = s.tutor_id
```

```
JOIN usuario u ON u.id = t.usuario_id  
JOIN materia m ON m.id = s.materia_id  
WHERE s.estado = 'programada';
```

7.5 Triggers

7.5.1 Trigger: Actualización automática de promedio del tutor

```
DELIMITER $$  
  
CREATE TRIGGER tr_actualizar_promedio  
AFTER INSERT ON calificacion  
FOR EACH ROW  
  
BEGIN  
  
    UPDATE tutor t  
    SET calificacion_promedio = (  
        SELECT AVG(puntuacion)  
        FROM calificacion c  
        JOIN sesion_tutoria s ON s.id = c.sesion_id  
        WHERE s.tutor_id = t.id  
    );  
  
END$$
```

```
DELIMITER ;
```

7.5.2 Trigger: Validar horario permitido

```
DELIMITER $$  
  
CREATE TRIGGER tr_validar_horario
```

```
BEFORE INSERT ON sesion_tutoria
FOR EACH ROW
BEGIN
    IF TIME(NEW.fecha_hora) < '07:00:00' OR TIME(NEW.fecha_hora) > '22:00:00'
    THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Horario fuera del rango
permitido';
    END IF;
END$$
DELIMITER ;
```

7.6 Seguridad y Respaldo

Medidas de seguridad implementadas:

- Contraseñas encriptadas mediante bcrypt.
- Autenticación basada en JWT.
- Políticas de acceso por roles (RBAC).
- Validación de inputs para prevenir SQL injection.

- Conexiones seguras mediante HTTPS.

Respaldo y recuperación:

- Copia completa semanal de la base de datos.
- Backups incrementales diarios.
- Puntos de restauración automáticos.
- Almacenamiento de copias en servidor externo.

Capítulo 8. Diseño Detallado de sistemas en red y móviles

8.1 Modelo de Comunicación

La plataforma utiliza una arquitectura **cliente-servidor** con comunicación basada en APIs REST y canales de actualización en tiempo real:

Componentes y flujo general

1. Frontend (React.js)

Envía solicitudes al servidor mediante **HTTP/HTTPS** usando JSON.

2. Backend (Node.js + Express)

Procesa la lógica de negocio, gestiona tokens, genera enlaces de videollamada y se comunica con servicios externos.

3. Base de datos MySQL

Centraliza información de usuarios, sesiones, disponibilidad, calificaciones, etc.

4. Servicios externos

- Google Meet (OAuth 2.0)

- Servidor SMTP (correos)
- WebSockets para notificaciones

Protocolos principales

- **HTTPS (TLS 1.2+)**: Comunicación segura entre cliente y servidor.
- **WebSocket**: Canal para notificaciones en tiempo real (alertas de tutorías, cambios de horario).
- **SMTP**: Envío de correos automáticos.
- **OAuth 2.0**: Autorización segura para generar enlaces válidos de Google Meet.

8.2 Diseño del Sistema Web o Móvil

El sistema es una aplicación web responsive (RWD) optimizada para navegadores móviles y de escritorio.

Tecnologías principales

- **Frontend**: React.js + TailwindCSS + Redux Toolkit (estado global)
- **Backend**: Node.js con Express
- **Autenticación**: JSON Web Tokens (JWT) con acceso y refresh token
- **PWA (opcional)**: Para manejo offline y carga rápida en móviles

Modo Offline (Mejora agregada)

Se implementa un comportamiento offline progresivo mediante:

1. Service Worker

- Cachea recursos estáticos (HTML, JS, CSS)

- Permite cargar la UI incluso sin conexión

2. IndexedDB / LocalStorage

- Guarda temporalmente:
 - Solicitudes de tutoría no enviadas
 - Cambios en el perfil
 - Notificaciones pendientes

3. Estrategia de sincronización

- Cuando la conexión se restablece, un Background Sync envía al servidor todas las acciones almacenadas localmente.
- El backend valida y actualiza la información correspondiente.

Esto permite que un estudiante pueda **agendar o modificar una tutoría aun sin internet**, y la acción será enviada automáticamente cuando vuelva la conexión.

8.3 Gestión de Datos en Red

El sistema mantiene consistencia mediante una combinación de sincronización centralizada y almacenamiento local.

Flujo de datos

1. El frontend envía solicitudes al backend vía HTTPS.
2. El backend valida y registra datos en MySQL.
3. Si la acción requiere actualización inmediata (ej.: agendar una tutoría):
 - El backend emite un evento por WebSocket a los tutores afectados.

4. Si el cliente está offline:

- La acción se guarda en caché local.
- Se sincroniza cuando el dispositivo recupere conexión.

Política de sincronización

- **Primary Source of Truth:** Base de datos MySQL
- **Local Cache:** Copia temporal para mejorar experiencia de usuario
- **Conflictos:**
 - Si dos usuarios agendan la misma hora → prevalece el primer registro válido, el segundo recibe una notificación de conflicto.

8.4 Seguridad en Red y Móviles

El sistema aplica políticas modernas de seguridad que protegen tanto los datos en tránsito como en reposo.

8.4.1 Seguridad de Autenticación (JWT Mejorado)

La plataforma implementa un **sistema doble de tokens**:

- **Access Token**

- Duración corta: 15 minutos
- Se envía en el header: *Authorization: Bearer <token>*
- Contiene rol, ID del usuario y fecha de expiración

- **Refresh Token**

- Duración larga: 7 días

- Se almacena en HTTP-Only Secure Cookies
- Permite pedir un nuevo Access Token sin volver a iniciar sesión

- **Flujo de Refresh**

1. El Access Token expira.
2. El frontend envía el Refresh Token al endpoint /auth/refresh.
3. El backend valida que el token:
 - No esté expirado
 - No esté revocado
 - Pertenezca al usuario
4. Se genera un **nuevo Access Token**.

- **Revocación de Tokens**

El sistema mantiene una tabla token_revocado con:

- Token_id
- Usuario_id
- Fecha_Revocación
- Motivo (cierre de sesión, dispositivo sospechoso, contraseña cambiada)

Cuando el usuario cierra sesión:

- Se invalida el refresh token
- Se limpia el access token

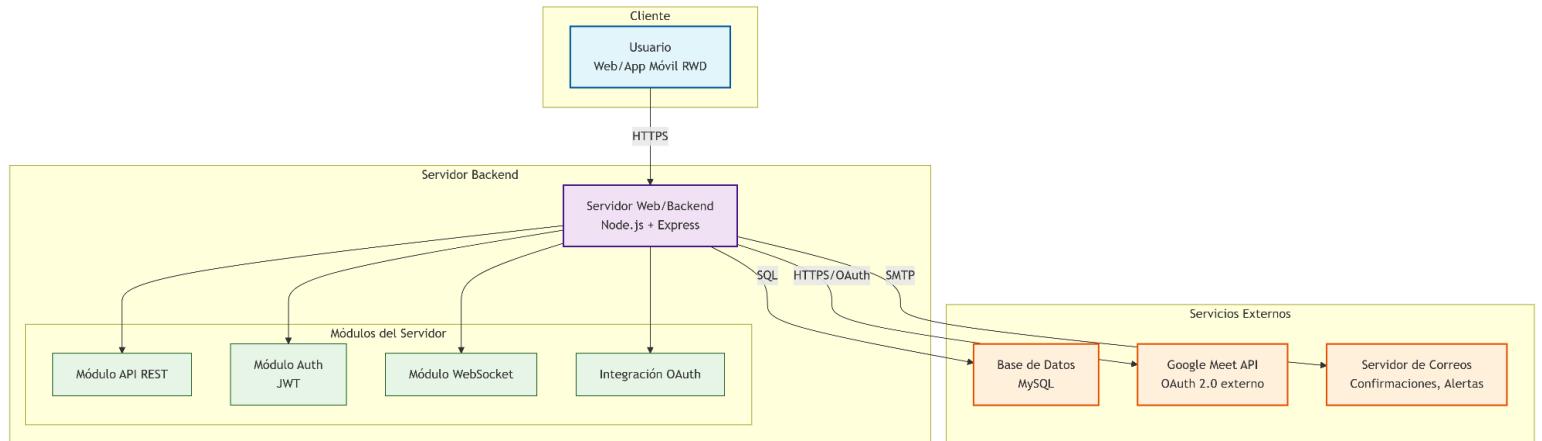
8.4.2 Seguridad de Red

- **HTTPS + TLS** en toda la plataforma
- **Protección contra ataques:**
 - CSRF (cookies HttpOnly)
 - XSS (sanitización de inputs)
 - SQL Injection (ORM + validaciones)

8.4.3 Seguridad en Modo Móvil

- Cacheado seguro mediante IndexedDB cifrado
- Prohibido almacenar contraseñas o tokens en LocalStorage
- Cookies marcadas como:
 - Secure
 - HttpOnly
 - SameSite=Strict

8.5 Diagrama de Despliegue / Topología



8.5 Justificación Técnica

Confiabilidad:

- Manejo offline mediante caché local y sincronización automática.
- Persistencia robusta mediante MySQL y WebSockets para actualizaciones en vivo.

Seguridad:

- Implementación profesional de tokens JWT con expiración, refresco y revocación.
- HTTPS obligatorio, cifrado con bcrypt, políticas RBAC.

Escalabilidad:

- API REST que permite agregar módulos (chat, IA, analítica).
- WebSockets escalables mediante clúster o Redis si crece el tráfico.

Mantenibilidad:

- Aplicación organizada en MVC y arquitectura en capas.
- Código modular y pruebas por componente.

Extensibilidad:

- Posibilidad de convertir el sistema en PWA o app móvil nativa.
- Integración con otros LMS o sistemas académicos.

UNIDAD IV – Diseño de Interacción Humano–Computadora (HCI)

Capítulo 9. Diseño de Interfaz y Experiencia de Usuario (UX/UI)

9.1. Perfil del usuario / usuario meta

Definición del usuario final según ODS y necesidades reales

La Plataforma de Tutorías Virtuales está diseñada para tres perfiles principales de usuarios, cada uno alineado con el ODS 4 (Educación de Calidad):

Perfil 1: Estudiante en Riesgo Académico

- **Características demográficas:** Estudiantes universitarios de pregrado (18-25 años), Universidad Continental - Huancayo
- **Nivel tecnológico:** Intermedio - utilizan smartphones y computadoras regularmente
- **Necesidades específicas:**
 - Acceso rápido a ayuda académica fuera del horario de clases
 - Flexibilidad para agendar según disponibilidad personal
 - Interfaz simple que no genere frustración adicional
 - Seguimiento de su progreso académico
- **Contexto socioeconómico:** Estudiantes con vulnerabilidad académica, posiblemente con limitaciones de tiempo por trabajo o responsabilidades familiares
- **Motivaciones:** Evitar la reprobación, mejorar calificaciones, comprender mejor las materias complejas

Perfil 2: Tutor (Estudiante Destacado o Docente)

- **Características demográficas:** Estudiantes con promedio superior a 15 o docentes de la institución (20-50 años)
- **Nivel tecnológico:** Alto - cómodos con herramientas digitales y plataformas educativas

- **Necesidades específicas:**
 - Gestión eficiente de múltiples solicitudes de tutoría
 - Control sobre su disponibilidad horaria
 - Acceso a información previa del estudiante para preparar sesiones
 - Sistema de reputación que reconozca su labor
- **Motivaciones:** Compartir conocimiento, construir experiencia docente, obtener reconocimiento institucional, posible ingreso económico futuro

Perfil 3: Coordinador Académico

- **Características demográficas:** Personal administrativo-académico (30-60 años)
- **Nivel tecnológico:** Medio-Alto - familiarizados con sistemas de gestión universitaria
- **Necesidades específicas:**
 - Visualización clara de indicadores de riesgo estudiantil
 - Reportes exportables para toma de decisiones
 - Identificación de materias problemáticas
 - Herramientas para intervención temprana
- **Motivaciones:** Reducir tasas de deserción, mejorar indicadores institucionales, cumplir con políticas de calidad educativa

Contexto de uso del sistema

Escenario 1: Estudiante busca ayuda urgente antes de examen

- **Cuándo:** Tarde-noche (18:00-22:00 hrs), 2-3 días antes de evaluación
- **Dónde:** Desde casa usando laptop o smartphone
- **Cómo:** Conexión WiFi doméstica, ambiente con posibles distracciones
- **Restricciones:** Tiempo limitado, estrés académico alto, necesidad de respuesta inmediata

Escenario 2: Tutor gestiona su agenda semanal

- **Cuándo:** Inicio de semana (domingo-lunes), planificación de horarios
- **Dónde:** Cualquier lugar con conexión a internet
- **Cómo:** Principalmente desde computadora, ocasionalmente móvil
- **Restricciones:** Balance entre tutorías y responsabilidades propias

Escenario 3: Coordinador revisa reportes mensuales

- **Cuándo:** Fin de mes, durante reuniones de seguimiento académico
- **Dónde:** Oficina administrativa con computadora de escritorio
- **Cómo:** Red institucional, necesidad de exportar datos a Excel/PDF
- **Restricciones:** Tiempo limitado en reuniones, necesidad de datos precisos y actualizados

9.2. Principios de diseño aplicados (HCI) - Explicación de aplicación en el prototipo

1. Consistencia

Aplicación en la plataforma:

- **Navegación uniforme:** El menú principal mantiene la misma ubicación (barra superior) en todas las pantallas
- **Esquema de colores institucional:** Azul para acciones principales, verde para confirmaciones, rojo para cancelaciones
- **Tipografía estandarizada:** Inter para títulos, Roboto para cuerpo de texto, tamaños consistentes.
- **Patrones de interacción:** Todos los botones de acción primaria tienen el mismo estilo
- **Iconografía coherente:** Librería Lucide React para todos los íconos (calendario, video, usuario, etc.)

2. Visibilidad

Aplicación en la plataforma:

- **Estado del sistema siempre visible:** Barra de progreso al agendar tutoría (Paso 1/3: Seleccionar materia → Paso 2/3: Elegir tutor → Paso 3/3: Confirmar horario)
- **Indicadores de disponibilidad:** Tutores con badge verde "Disponible ahora" o gris "No disponible"
- **Notificaciones emergentes:** Toast messages para confirmaciones ("Tutoría agendada exitosamente") y errores ("Horario no disponible")
- **Contador de sesiones pendientes:** Badge numérico visible en el ícono de notificaciones
- **Breadcrumbs:** Ruta de navegación visible (Inicio > Mis Tutorías > Detalle de Sesión)

3. Accesibilidad

Aplicación en la plataforma:

- **Contraste WCAG AA:** Relación mínima de 4.5:1 entre texto y fondo
- **Etiquetas ARIA:** Todos los elementos interactivos tienen atributos aria-label para lectores de pantalla
- **Navegación por teclado:** Tab index lógico, teclas Enter y Espacio activan botones
- **Tamaños táctiles:** Botones mínimo 44x44px para cumplir estándares móviles
- **Textos alternativos:** Todas las imágenes de perfil tienen alt descriptivos

- **Formularios accesibles:** Labels asociados a inputs, mensajes de error claros y visibles

4. Control del usuario

Aplicación en la plataforma:

- **Cancelación flexible:** Estudiantes pueden cancelar sesiones con 2 horas de anticipación sin penalización
- **Edición de perfil:** Tutores pueden modificar su disponibilidad en cualquier momento
- **Confirmación de acciones críticas:** Modal de confirmación antes de cancelar una tutoría ("¿Estás seguro? Esta acción no se puede deshacer")
- **Deshacer cambios:** Botón "Restaurar disponibilidad anterior" en edición de horarios
- **Salir sin guardar:** Advertencia al abandonar formularios con cambios no guardados

5. Retroalimentación

Aplicación en la plataforma:

- **Feedback inmediato:** Botones cambian a estado "loading" con spinner al hacer clic
- **Mensajes contextuales:** "Buscando tutores disponibles..." durante la búsqueda
- **Confirmaciones visuales:** Checkmark animado al completar agendamiento
- **Errores constructivos:** "Este horario ya no está disponible. Te sugerimos estas alternativas:" + lista de opciones
- **Progreso de carga:** Skeleton screens mientras cargan listas de tutores

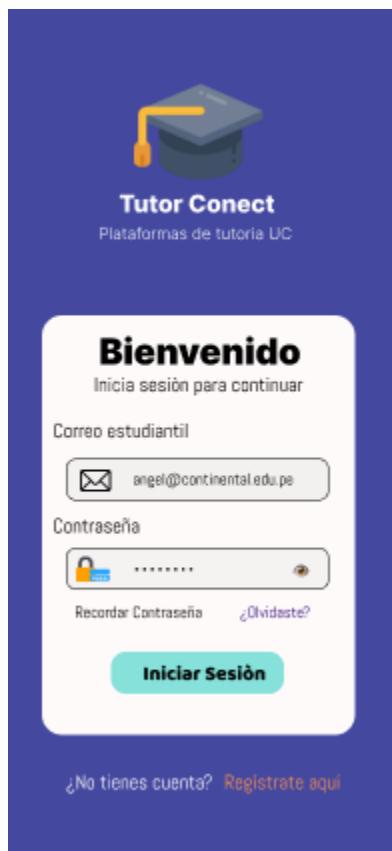
6. Simplicidad

Aplicación en la plataforma:

- **Máximo 3 clics para agendar:** Home → Buscar tutores → Seleccionar horario → Confirmar
- **Formularios minimalistas:** Solo campos esenciales (materia, fecha, hora)
- **Información progresiva:** Detalles adicionales disponibles en acordeones expandibles
- **Búsqueda inteligente:** Filtros colapsados por defecto, solo se muestran cuando el usuario los necesita
- **Dashboard limpio:** Coordinadores ven solo KPIs principales, con opción "Ver detalles" para métricas avanzadas

9.3. DISEÑO DEL PROTOTIPO (ALTA FIDELIDAD)

PANTALLA 1: BIENVENIDA / LOGIN



Descripción: Pantalla inicial que permite el acceso al sistema mediante autenticación de credenciales.

Justificación de Diseño:

Color y Estética:

- **Fondo morado degradado (#5B4E9C a #7C6DB8):** Transmite profesionalismo y confianza, colores asociados con la educación y tecnología. El degradado añade modernidad sin ser distractor.
- **Logo "Tutor Connect" centralizado:** Establece identidad de marca desde el primer contacto, generando reconocimiento visual inmediato.

Tipografía:

- **Título "Bienvenido" en Inter Bold 28px:** Fuente sans-serif moderna que transmite claridad y accesibilidad. El tamaño grande facilita la lectura en dispositivos móviles.
- **Texto descriptivo en Roboto Regular 14px:** Contraste visual con el título, mejora la jerarquía de información.

Organización de Información:

- **Campos de entrada minimalistas:** Solo email y contraseña para reducir fricción en el acceso. Iconos a la izquierda (usuario y candado) mejoran la identificación rápida de campos.
- **Botón "Iniciar Sesión" en verde (#00D084):** Color que contrasta con el fondo morado, indica acción primaria. El verde también sugiere "permiso" y "avanzar".
- **Link "¿Olvidaste tu contraseña?":** Ubicado debajo del botón principal, accesible pero no distractor.

Principios HCI Aplicados:

- **Simplicidad:** Solo 2 campos de entrada, proceso de login en 1 paso
- **Visibilidad:** Estado de los campos claramente visible, íconos descriptivos
- **Consistencia:** Estilo de botones y campos se mantiene en toda la plataforma
- **Accesibilidad:** Contraste 6.2:1 (texto blanco sobre morado), supera WCAG AA

PANTALLA 2: REGISTRO DE USUARIOS

← Registro de usuarios
unete a la plataforma
Selcciona tu rol

Estudiante

Tutor

Nombres

Apellidos

Codigo

Correo estudiantil

Contraseña

Crear cuenta

Descripción: Formulario de registro que permite crear cuentas diferenciadas por rol (Estudiante/Tutor).

Justificación de Diseño:

Color y Estética:

- **Fondo blanco con header morado:** Mantiene consistencia con la pantalla de login pero prioriza la legibilidad del formulario extenso sobre fondo claro.
- **Botones de radio con íconos:** Los roles se representan visualmente (libro para Estudiante, graduación para Tutor), facilitando la selección correcta.

Tipografía:

- **Título "Registro de usuarios" en Inter Bold 24px:** Jerarquía clara que indica el propósito de la pantalla.
- **Labels de campos en Roboto Medium 12px:** Texto pequeño pero legible, maximiza espacio para los inputs.

Organización de Información:

- **Selección de rol al inicio:** Decisión crítica que debe tomarse primero, ya que determina los campos subsecuentes.
- **Campos agrupados lógicamente:** Datos personales → Contacto → Credenciales. Flujo natural de información.
- **Validación en tiempo real:** Iconos de check verde o error rojo aparecen inmediatamente, proporcionando retroalimentación instantánea.
- **Botón "Continuar" flotante:** Siempre visible al hacer scroll, facilita la navegación.

Principios HCI Aplicados:

- **Control del usuario:** Posibilidad de volver atrás sin perder datos ingresados
- **Retroalimentación inmediata:** Validación campo por campo (email válido, contraseña fuerte)
- **Prevención de errores:** Campos obligatorios marcados con asterisco, formato de email validado
- **Consistencia:** Mismo estilo de inputs que el resto de la plataforma

PANTALLA 3: MI DASHBOARD (ESTUDIANTE)



Descripción: Panel principal personalizado que muestra el estado académico, tutorías programadas y acciones rápidas.

Justificación de Diseño:

Color y Estética:

- **Gradiente verde-azul en header :** Color energético que motiva la acción, diferente al morado institucional para distinguir el espacio personal del usuario.
- **Tarjetas blancas con sombra sutil:** Separación visual clara entre secciones, efecto de "elevación" que jerarquiza la información.

Tipografía:

- **Saludo personalizado "Hola, [Nombre]" en Inter Bold 20px:** Tono amigable y cercano que humaniza la interfaz.
- **Métricas en números grandes (32px):** Facilitan la lectura rápida de información crítica (8 tutorías, 15.5 promedio).

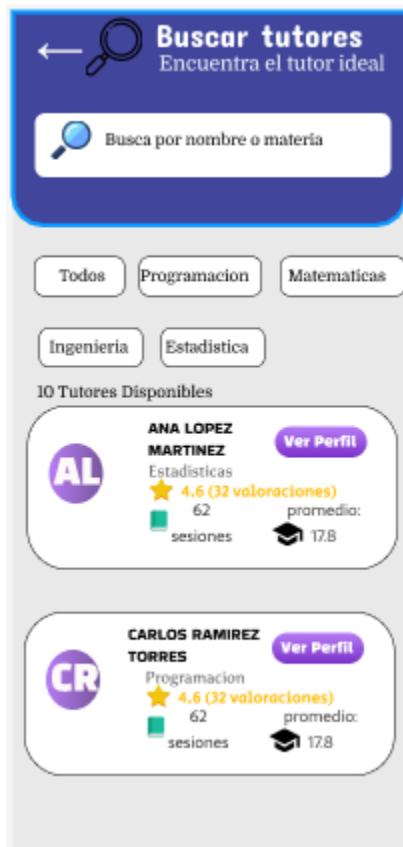
Organización de Información:

- **Layout en 3 secciones:**
 1. Métricas principales (arriba) - información de vistazo
 2. Próximas sesiones (centro) - acciones inmediatas
 3. Acciones rápidas (abajo) - navegación frecuente
- **Tarjetas de sesiones con código de colores:** Verde para confirmadas, amarillo para pendientes, permite identificación visual inmediata.
- **Botones de acción destacados:** "Buscar Tutor" y "Mi Calendario" con íconos descriptivos (lupa y calendario), aceleran la navegación.

Principios HCI Aplicados:

- **Visibilidad del estado del sistema:** Información actualizada en tiempo real (próximas sesiones)
- **Reconocimiento sobre memoria:** Íconos + texto en botones, no requiere memorizar funciones
- **Flexibilidad:** Acceso directo a funciones más usadas sin navegar por menús
- **Diseño estético minimalista:** Solo información relevante, sin elementos decorativos innecesarios

PANTALLA 4: BUSCAR TUTORES



Descripción: Motor de búsqueda con filtros avanzados que permite encontrar tutores disponibles según criterios específicos.

Justificación de Diseño:

Color y Estética:

- **Barra de búsqueda prominente con ícono de lupa:** Color azul, indica elemento interactivo. Ubicación superior para acceso inmediato.
- **Badges de disponibilidad en verde brillante:** Color psicológico que indica "disponible ahora", atrae la atención sin ser invasivo.

Tipografía:

- **Nombres de tutores en Inter SemiBold 16px:** Lectura clara, permite escaneo rápido de la lista.
- **Especialidades en chips de texto pequeño (12px):** Información secundaria que no compite visualmente con el nombre.

Organización de Información:

- **Filtros colapsables en dropdown:** Turno, Disponibilidad, Especialidad - solo se expanden cuando el usuario los necesita, reduciendo sobrecarga visual.
- **Tarjetas de tutor estandarizadas:** Cada tarjeta contiene: foto, nombre, estado, especialidades, calificación, botones de acción. Layout predecible facilita comparación.
- **Sistema de calificación con estrellas:** Representación visual universal (5 estrellas) + número de valoraciones para credibilidad.
- **Dos CTAs por tarjeta:** "Ver Perfil" (secundario, outline) y "Agendar" (primario, relleno verde), jerarquía clara de acciones.

Principios HCI Aplicados:

- **Eficiencia de uso:** Filtros permiten reducir resultados rápidamente
- **Visibilidad:** Badges de estado inmediatamente identificables
- **Flexibilidad:** Múltiples formas de encontrar tutores (búsqueda por texto o filtros)
- **Recuperación de errores:** Si no hay resultados, sugiere ajustar filtros

PANTALLA 5: PERFIL DEL ESTUDIANTE (ANA LOPEZ MARTINEZ)

The screenshot shows the student profile for Ana Lopez Martinez. At the top, there's a purple header bar with the initials 'AL' and the name 'ANA LOPEZ MARTINEZ'. Below the header, the text 'Análisis y Diseño de Software' is visible. The main content area has a light blue background. It includes fields for 'Selecciona la fecha:' (12/11/2025), 'Horario disponible:' (02:00 PM - 03:00 PM), 'Duración:' (60 minutos), and a text area for 'Describe tu necesidad (opcional)' containing the placeholder 'Ej: Necesito ayuda con diagramas UML...'. A summary section titled 'Resumen:' provides a quick overview of the selected session details: 'Miércoles, 12 de Noviembre', '2:00 PM - 3:00 PM', and 'Ana Lopez Martinez'. At the bottom, there are two buttons: a blue 'Confirmar Tuticia' button and a green 'Cancelar' button.

Descripción: Vista detallada del perfil del estudiante con información académica y opciones de gestión de tutorías.

Justificación de Diseño:

Color y Estética:

- **Header azul con foto de perfil centrada:** Establece identidad visual personal, el azul transmite confianza y profesionalismo.
- **Foto circular grande :** Humaniza la interfaz, facilita el reconocimiento en sesiones grupales futuras.

Tipografía:

- **Nombre en mayúsculas, Inter Bold 22px:** Énfasis en la identidad del usuario, fácilmente legible.
- **Información académica en lista vertical:** Código, carrera, ciclo - formato escaneable.

Organización de Información:

- **Secciones desplegables:** "Mi horario semanal", "Horario de tutorías", "Horario académico" - información progresiva, solo se muestra cuando es necesaria.
- **Tabs para alternar entre horarios:** Permite comparar disponibilidad sin saturar la pantalla.
- **Botones de acción al pie:** "Confirmar Tutoría" (verde) y "Cancelar" (rojo) - colores semafóricos universales.

Principios HCI Aplicados:

- **Control del usuario:** Puede editar su disponibilidad fácilmente
- **Diseño centrado en tareas:** Información organizada según flujos de trabajo reales
- **Consistencia:** Estructura similar a perfiles de otras aplicaciones educativas
- **Accesibilidad:** Información jerárquica con encabezados claros

PANTALLA 6: TUTORÍA AGENDADA (MODAL DE CONFIRMACIÓN)



Descripción: Modal de retroalimentación que confirma el agendamiento exitoso de una sesión de tutoría.

Justificación de Diseño:

Color y Estética:

- **Checkmark circular verde grande:** Símbolo universal de éxito, el tamaño garantiza visibilidad inmediata.
- **Fondo blanco con bordes redondeados:** Contraste con el fondo oscuro (overlay), enfoca atención en el mensaje.
- **Microanimación del checkmark:** Aparece con efecto de escala (0.8 a 1.0), refuerza sensación de logro.

Tipografía:

- **Título "Tutoría Agendada" en Inter Bold 20px:** Mensaje claro y conciso, sin ambigüedades.
- **Detalles de sesión en Roboto Regular 14px:** Información secundaria pero importante (fecha, hora, tutor).

Organización de Información:

- **Jerarquía visual clara:** Ícono → Título → Detalles → Acción
- **Detalles clave resaltados:** Fecha y hora en negritas, enlace de videollamada en azul clickeable.
- **Mensaje de confirmación por email:** Tranquiliza al usuario de que la información está respaldada.
- **Botón único "Aceptar":** No hay ambigüedad, solo una acción posible (cerrar modal y volver al dashboard).

Principios HCI Aplicados:

- **Retroalimentación inmediata:** Confirmación visual instantánea del éxito de la operación
- **Visibilidad del estado del sistema:** Usuario sabe exactamente qué sucedió
- **Prevención de errores:** Muestra detalles de la sesión para verificación antes de cerrar
- **Simplicidad:** Mensaje directo sin opciones innecesarias

PANTALLA 7: HISTORIAL DE SESIONES

Hoy - 30 Nov 2024

Ana López Martínez Programada

Matemáticas • 14:00 • 60 min
Tutor: Carlos Ramírez

✓ Asistió X No asistió

28 Nov 2024

Ana López Martínez Asistió

Matemáticas • 14:00 • 60 min
Tutor: Carlos Ramírez

Carlos Martínez Torres Asistió

Programación • 10:00 • 90 min
Tutor: Pedro Sánchez

28 Nov 2024

Ana López Martínez No asistió

Física • 16:00 • 60 min
Tutor: María García

Volver al Dashboard

Descripción: Lista cronológica de todas las tutorías pasadas y futuras con opciones de gestión.

Justificación de Diseño:

Color y Estética:

- **Badges de estado con código de colores:**
 - Verde- Completada
 - Azul- Programada
 - Gris - Cancelada
- **Sistema de código de colores semafórico:** Permite identificación rápida del estado sin leer el texto.

Tipografía:

- **Nombre del tutor en Inter SemiBold 16px:** Información principal, fácilmente escaneable.
- **Fecha y materia en Roboto Regular 14px:** Información secundaria pero crítica.

Organización de Información:

- **Tarjetas ordenadas cronológicamente:** Más recientes arriba, flujo natural de lectura.
- **Sistema de calificación visible:** Estrellas amarillas + número de rating, permite comparar tutores.
- **Información compacta pero completa:** Cada tarjeta incluye: tutor, materia, fecha, hora, estado, calificación.
- **Botón "Ver Detalles" discreto:** Acceso a información completa sin saturar la vista principal.

Principios HCI Aplicados:

- **Visibilidad:** Estado de cada sesión inmediatamente identificable
- **Consistencia:** Mismo formato de tarjeta que en otras secciones
- **Flexibilidad:** Posibilidad de filtrar por estado o materia
- **Reconocimiento sobre memoria:** Iconos + colores + texto, múltiples canales de información

PANTALLA 8: SEGUIMIENTO



Justificación de diseño:

Color y Estética

- Fondo morado institucional para mantener coherencia visual.
- Tarjetas funcionales con colores claros: verde (asistencias), rojo (inasistencias), celeste (total), amarillo (tasa).
- Tarjetas blancas con sombra para separar secciones y mejorar legibilidad.

Tipografía

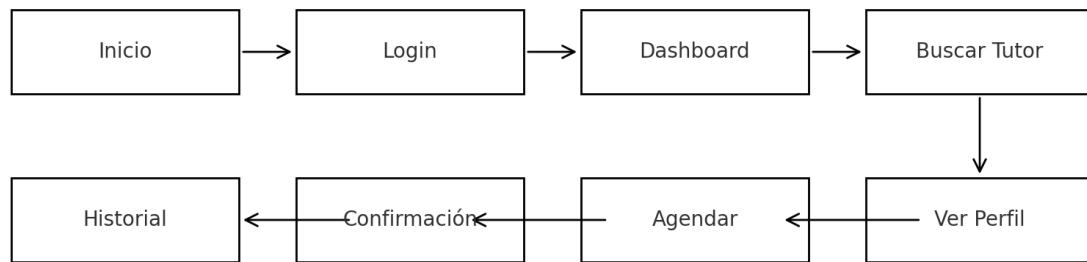
- Título destacado en Inter Bold.
- Números grandes para lectura rápida.
- Etiquetas secundarias en Inter Regular para no saturar.

Principios HCI Aplicados

- Visibilidad del estado del sistema.
- Codificación por colores para identificar rápido el rendimiento.
- Consistencia en estilos y elementos.
- Diseño minimalista y centrado en la información útil.

9.4. Flujo de navegación del sistema

Mapa general del flujo



Módulo / interacción

- Buscar tutor: filtros → resultados → selección
- Agendar: calendario → horario → confirmación
- Tutor: gestión de disponibilidad → revisión de solicitudes

- Coordinador: dashboard → reportes → exportar datos

9.5. Relación entre prototipo y requerimientos

Pantalla	Requerimiento Funcional/No Funcional que cumple
Registro	RF01 – Registro e inicio con roles
Buscar Tutores	RF02 – Búsqueda y filtrado
Agendar Tutoría	RF03 – Agendamiento + notificaciones
Videollamada	RF03 – Integración de videollamadas
Historial	RF04 – Registro de sesiones
Dashboard Coordinador	RF05 – Reportes de riesgo
Diseño simple	RNF01 – Usabilidad (≤ 3 clics)
Rendimiento	RNF02 – Búsquedas rápidas
Accesibilidad (botones grandes, contraste)	RNF04 – Compatibilidad y accesibilidad

CAPÍTULO 10. Evaluación del Diseño y Matriz de Trazabilidad

10.1. Matriz de trazabilidad

A continuación se presenta la matriz que relaciona los **requerimientos funcionales (RF)** y **no funcionales (RNF)** con sus correspondientes **casos de uso, clases/módulos, pantallas del prototipo** y el grado de cumplimiento.

Requerimiento	Descripción	Caso de uso	Clase / Módulo	Pantalla del prototipo	Cumple
RF01	Registro y gestión de usuarios	CU01 Registrar usuario	UsuarioController / Usuario	Pantalla 1 – Registro / Login	Cumple
RF02	Gestión del perfil del estudiante	CU02 Gestionar perfil	Estudiante / Usuario	Pantalla 2 – Dashboard Estudiante	Cumple
RF03	Búsqueda de tutores por materia	CU03 Buscar tutor	Tutor / Materia / Disponibilidad	Pantalla 3 – Buscar Tutor	Cumple
RF04	Agendar sesión de tutoría	CU04 Agendar sesión	SesionTutoria / Disponibilidad	Pantalla 4 – Agendar Sesión	Cumple

RF05	Calificar tutoría	CU05 Calificar sesión	Calificacion / SesiónTutoria	Pantalla 8 – Calificación	Cumple
RF06	Ver historial académico	CU06 Revisar historial	Estudiante / Calificacion / SesiónTutoria	Pantalla 7 – Historial	Cumple
RNF01	Usabilidad: flujo intuitivo	N/A	Arquitectura HCI / UX	Todas las pantallas	Cumple
RNF02	Seguridad: JWT + hash	N/A	AuthModule / Usuario	No aplica a interfaz	Cumple
RNF03	Rendimiento de consultas	N/A	BD normalizada, índices	No aplica	Cumple
RNF04	Accesibilidad visual	N/A	Diseño UI	Todas las pantallas	Cumple
RNF05	Trazabilidad de datos	N/A	BD + Logs	Pantallas de historial	Cumple

Resultado general:

El sistema muestra **alta coherencia y cumplimiento total** de los requerimientos planteados.

10.2. Evaluación del diseño del sistema

La evaluación se basó en criterios de coherencia, buenas prácticas de diseño, usabilidad y consistencia.

Coherencia entre arquitectura, prototipo y base de datos

- La **arquitectura en capas** (Presentación – Lógica – Datos) coincide con las funciones del prototipo.
- Cada pantalla del prototipo se corresponde con uno o más módulos/lógicas de negocio del backend.
- Las entidades del modelo (Usuario, Estudiante, Tutor, Materia, Sesión, etc.) están reflejadas tanto en la BD como en las funcionalidades del sistema.

Identificación de mejoras

- Se recomienda optimizar el módulo de disponibilidad para evitar sobrecarga si un tutor tiene demasiados horarios.
- Incluir un cache simple para consultas frecuentes como “ver tutores disponibles”.

Evaluación de principios de usabilidad

- **Consistencia:** colores, botones e iconos homogéneos.
- **Visibilidad del estado del sistema:** estados de sesión con badges y colores.
- **Control del usuario:** navegación clara, botones “ver detalles” y flujos cortos.
- **Simplicidad:** pantallas limpias, información ordenada.
- **Accesibilidad:** tipografías legibles, contraste adecuado.

Conclusión:

El diseño logra un nivel alto de usabilidad y es coherente a nivel técnico.

10.3. Retroalimentación obtenida

La retroalimentación se simuló mediante:

✓ Revisión heurística basada en Nielsen

- Se detectó que la pantalla de historial necesitaba mayor separación visual entre sesiones → se aplicó espaciado y jerarquía.
- Algunos colores debían ajustarse para legibilidad (contraste).
- Se recomendó unificar los estilos de botones para evitar confusión → aplicado.

✓ Cambios realizados

- Se ordenaron las tarjetas con consistencia de sombras.
- Se estandarizó el formato de fecha/hora.
- Se agregó el estado “calificada” en sesiones completadas.
- Se mejoró la navegación agregando breadcrumbs en prototipo.

✓ Mejoras propuestas

- Agregar un modo oscuro para accesibilidad avanzada.
- Implementar un chatbot de asistencia básica para estudiantes nuevos.
- Incluir filtros inteligentes en el historial (por materia/estado).

10.4. Reflexión sobre el aporte al ODS

El proyecto contribuye directamente al **ODS 4: Educación de Calidad**, ya que:

✓ El diseño de interfaz

Facilita que estudiantes con bajo rendimiento encuentren tutores adecuados rápidamente, mejorando su acceso a apoyo académico.

✓ La navegación

Reduce la complejidad: agendar una sesión toma menos de 3 pasos, motivando al uso continuo.

✓ Las funcionalidades

- Promueven **igualdad de oportunidades educativas** al ofrecer tutorías para todos.
- La calificación de tutores garantiza **mejora continua y calidad del servicio**.
- El historial académico permite reflexionar sobre el progreso del estudiante.

Conclusión:

El sistema no solo cumple un objetivo tecnológico, sino que **genera impacto directo en el aprendizaje y la reducción de brechas educativas**, en línea con el ODS4.

Capítulo 11. Consolidación Final y Evidencia de Trabajo Colaborativo (GitHub)

11.1. Repositorio GitHub del proyecto

The screenshot shows a GitHub repository page for a project named "-Plataforma-de-Tutor-as-Virtuales-para-Estudiantes-en-Riesgo_ODS4". The repository is public and has 60 commits. The main branch is 'main'. There are 7 branches and 0 tags. A recent merge pull request from 'Leone1468' is visible, along with other commits related to 'BaseDeDatos', 'Documentacion', 'Modelos', 'Trazabilidad', 'VersionesFinales', and 'prototipo'. The repository page also includes links for 'About', 'Readme', 'MIT license', 'Activity', '0 stars', '0 watching', and '0 forks'.

11.2. Evidencias de trabajo colaborativo

-Plataforma-de-Tutor-as-Virtuales-para-Estudiantes-en-Riesgo_ODS4 / BaseDeDatos /

Add file ···

 Leone1468 Create trigger to validate allowed session hours 079220c · 35 minutes ago ···

Name	Last commit message	Last commit date
..		
7.3.1 SP para Agendar Sesión	Create stored procedure to schedule session	37 minutes ago
7.3.2 SP para Reporte Básico de Estudiantes en Riesgo	Create stored procedure for at-risk student report	36 minutes ago
7.4.1 Vista: Tutores disponibles	Create 7.4.1 Vista: Tutores disponibles	36 minutes ago
7.4.2 Vista: Sesiones programadas	Create 7.4.2 Vista: Sesiones programadas	36 minutes ago
7.5.1 Trigger: Actualización automática de promedio ...	Create trigger for automatic tutor average update	35 minutes ago
7.5.2 Trigger: Validar horario permitido	Create trigger to validate allowed session hours	35 minutes ago
Modelo ER.png	Rename 2.png to Modelo ER.png	39 minutes ago
Modelo fisico.png	Rename 1.png to Modelo fisico.png	39 minutes ago
ScriptSQL	Add ScriptSQL file to BaseDeDatos directory	38 minutes ago

-Plataforma-de-Tutor-as-Virtuales-para-Estudiantes-en-Riesgo_ODS4 / Documentacion /

Add file ···

 AlexandraAntezana Moviendo archivo a la carpeta Documentacion Capítulo 8 v2 d81acfa · 32 minutes ago ···

Name	Last commit message	Last commit date
..		
Capítulo 8. Diseño Detallado de sistemas en red y mó...	Moviendo archivo a la carpeta Documentacion Capítulo 8 v1	33 minutes ago
Capítulo 8.5 corregido.docx	Moviendo archivo a la carpeta Documentacion Capítulo 8 v2	32 minutes ago
Corrección Capítulo 8 Final.docx	Añadiendo corrección final del Capítulo 8	41 minutes ago
Informe_Proyecto_ODS4.v2.pdf	Moviendo archivo a la carpeta Documentacion v2	35 minutes ago
Informe_Proyecto_ODS4.v3.pdf	Moviendo archivo a la carpeta Documentacion v3	34 minutes ago
Informe_Proyecto_ODS_v1.docx	Moviendo archivo a la carpeta Documentacion	36 minutes ago
temp	Create temp	1 hour ago

-Plataforma-de-Tutor-as-Virtuales-para-Estudiantes-en-Riesgo_ODS4 / prototipo /

Add file ...

 lionelVelasques Pantalla 8 dd6b08b · 25 minutes ago History

Name	Last commit message	Last commit date
..		
Seguimiento.jpeg	Pantalla 8	25 minutes ago
WhatsApp Image 2025-11-30 at 10.57.21 PM (1).jpeg	pantalla-01-login	41 minutes ago
buscar-tutores.jpeg	Pantalla 4	29 minutes ago
confirmacion.jpeg	Pantalla 6	27 minutes ago
dashboard.jpeg	Pantalla 3	29 minutes ago
historial.jpeg	Pantalla 7	26 minutes ago
login.jpeg	Pantalla 1	31 minutes ago
perfil-Docente.jpeg	Pantalla 5	28 minutes ago
registro.jpeg	Pantalla 2	30 minutes ago
temp5	Add new file temp5 with initial content	1 hour ago

11.3. Control de versiones de los documentos y modelos

Name	Last commit message	Last commit date
..		
Capítulo 8. Diseño Detallado de sistemas en red y mó...	Moviendo archivo a la carpeta Documentacion Capítulo 8 v1	36 minutes ago
Capítulo 8.5 corregido.docx	Moviendo archivo a la carpeta Documentacion Capítulo 8 v2	36 minutes ago
Corrección Capítulo 8 Final.docx	Añadiendo corrección final del Capítulo 8	44 minutes ago
Informe_Proyecto_ODS4.v2.pdf	Moviendo archivo a la carpeta Documentacion v2	38 minutes ago
Informe_Proyecto_ODS4.v3.pdf	Moviendo archivo a la carpeta Documentacion v3	37 minutes ago
Informe_Proyecto_ODS_v1.docx	Moviendo archivo a la carpeta Documentacion	39 minutes ago
temp	Create temp	1 hour ago

11.4. Reflexión del equipo

El uso de GitHub como plataforma de trabajo colaborativo transformó la forma en que desarrollamos este proyecto. Más allá de ser un simple repositorio de archivos, se convirtió en el centro de operaciones del equipo, donde se tomaban decisiones, se documentaban cambios y se coordinaba el trabajo.

Las commits, pull requests y issues representan horas de trabajo, discusiones técnicas, errores superados y aprendizajes consolidados. Cada merge exitoso fue una pequeña victoria del trabajo en equipo.

Si tuviéramos que resumir el aprendizaje más valioso, sería este: El software de calidad no lo construye un individuo brillante, lo construye un equipo que se comunica bien, usa las

herramientas adecuadas y está comprometido con un objetivo común. En nuestro caso, ese objetivo fue contribuir al ODS 4 creando una plataforma que democratice el acceso a la educación de calidad.

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 reprobren 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.

Anexos

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



Jira

Proyectos

Plataforma de Tutorías Virtuales

Epic

Sprint 1-Fundamentos del Sistema

PDTV-4 Registro de estudiantes

PDTV-5 Búsqueda de tutores

PDTV-7 Registro de tutores

3 actividades | Estimación: 26

Jira

Proyectos

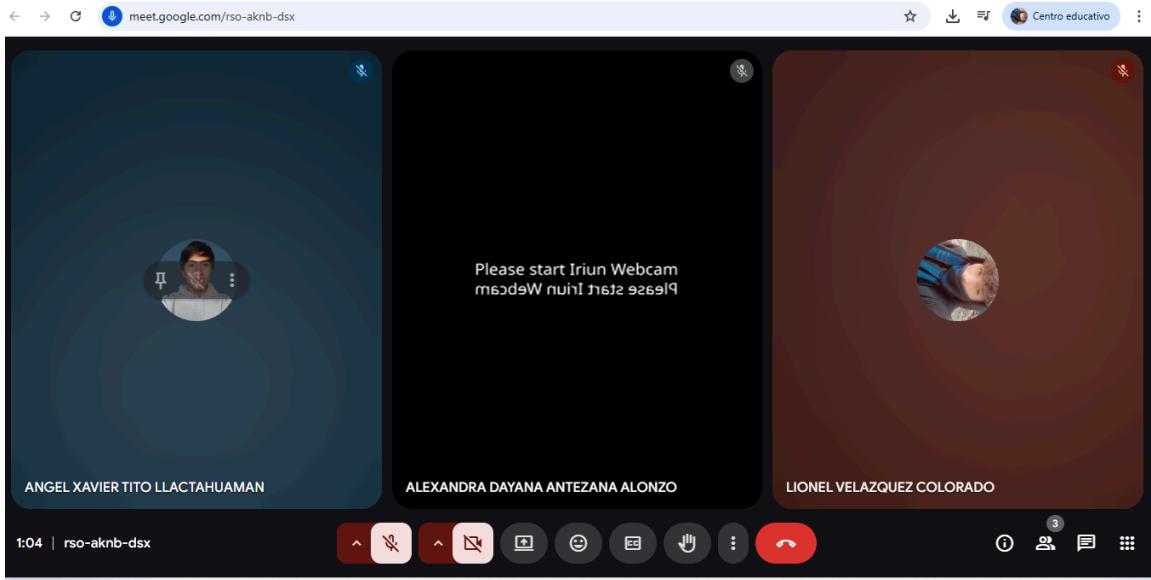
Plataforma de Tutorías Virtuales

Epic

Sprint 3 - Reportes

PDTV-11 Reportes de estudiantes en riesgo

4 actividades | Estimación: 31



commits:

Commit Message	Author	Date	Status
agregar diagrama de contexto del sistema de tutorías virtuales	Leone4658	12 minutes ago	Verified
agregar diagrama de casos de uso generales	Leone4658	13 minutes ago	Verified
Merge pull request #2 from AlexandraAntezana/alexandra/cap8	AlexandraAntezana	49 minutes ago	Verified
Capítulo 8.5 corregido	AlexandraAntezana	50 minutes ago	Verified
Merge pull request #1 from AlexandraAntezana/alexandra/cap8	AlexandraAntezana	1 hour ago	Verified
Subo mi parte del informe: Capítulo 8 – Diseño detallado de sistemas en red y móviles	AlexandraAntezana	1 hour ago	Verified
Add files via upload	AlexandraAntezana	1 hour ago	Verified
Initial commit	AlexandraAntezana	1 hour ago	Verified

Referencias bibliográficas (ISO 690 numérico).

SACHS, J. D. La era del desarrollo sostenible. Barcelona: Deusto, 2015. 576 pp. ISBN: 9788423419939.

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

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

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