



ESCUELA DE
INFORMÁTICA Y
TELECOMUNICACIONES

Educ-AR

(SAD) Software Architecture Document

Versión 1.2

Identificación de Documento

Identificación	DAS 1.2
Proyecto	Educ-AR
Versión	1.2

Documento mantenido por	Chiay Lin - Bruno Jimenez
Fecha de última revisión	02.10.2024
Fecha de próxima revisión	10.10.2024

Documento aprobado por	Josue Espinoza
Fecha de última aprobación	13.10.2024

Historia de Revisiones

Fecha	Versión	Descripción	Autor
02/10/2024	1.0	Creación de Diagramas y Generación de Documento DAS:	Educar
15/10/2024	1.1	Cambios en los modelos de clases, se han actualizado los modelos que se habían creado con anterioridad.	Educar
16/10/2024	1.2	Se afinan detalles del informe DAS:	Educar

Tabla de Contenidos

Introducción	3
1.1 Contexto del Problema	3
1.2 Propósito	4
1.3 Ámbito	4
1.4 Definiciones, acrónimos y abreviaciones	5
1.5 Referencias	6
1.5.1 Especificación de documentos	6
1.5.2 Resumen ejecutivo	6
1.6 Representación	7
1.6.1 Vista de Casos de Uso y Escenarios de Calidad	7
1.6.2 Vista de Metas y Restricciones	8
1.6.3.Vista Lógica	9
1.6.4 Vista de Procesos	16
2. Vista de Implementación	18
2.1 Diagrama de Componentes:	18
2.1 Diagrama de Paquetes:	18
3 Metas y Restricciones de la Arquitectura	19
3.1 Metas de la arquitectura	19
3.2 Restricciones de la Arquitectura	21
3.3 Otros antecedentes y consideraciones	22
4 Vista de Casos de Uso y Escenarios de Calidad	23
4.3 Modelo de Casos de Uso	23
4.2 Especificación de Casos de Uso Relevantes	23
4.3 Especificación de los Escenarios de Calidad Relevantes	24
5. Vista Lógica	26
5.1 Parte Estructural	26
5.2 Parte Dinámica	28
6. Vista de Procesos	32
7 Vista de Implementación	34
8. Vista de Despliegue	34
9. Decisiones de Diseño y Selección de Alternativas	36
10. Análisis de Reutilización	37

Introducción

A un grupo de estudiantes de DUOC UC se les ha pedido que desarrollen una aplicación de realidad aumentada y virtual para mejorar la formación de estudiantes de enfermería. El grupo ha decidido crear una simulación interactiva de triage, donde los usuarios podrán interactuar con pacientes virtuales y evaluar signos vitales, así como realizar un cuestionario para determinar el nivel de triage del paciente. El objetivo de esta aplicación es ofrecer un entorno de aprendizaje más inmersivo y práctico que prepare a los estudiantes para situaciones clínicas reales.

1.1 Contexto del Problema

El problema que se enfrenta es que, en el ámbito de la educación, especialmente en la formación de profesionales de la salud, la falta de recursos tecnológicos limita la calidad del aprendizaje. Actualmente, no existen sedes de DUOC UC que cuenten con un laboratorio de realidad aumentada en salud. Por esta razón, surge la necesidad de crear una solución innovadora que aproveche los avances tecnológicos, mejorando así la educación y ofreciendo a los estudiantes de la carrera de Técnico en Enfermería prácticas virtuales. Esto les permitirá familiarizarse con procedimientos y situaciones clínicas en un entorno seguro, de modo que, cuando enfrenten escenarios reales, no se sientan tan atemorizados y puedan actuar con confianza y efectividad.

1.2 Propósito

A un grupo de estudiantes de DUOC UC se les ha pedido que desarrollen una aplicación de realidad aumentada y virtual para mejorar la formación de estudiantes de enfermería. El grupo ha decidido crear una simulación interactiva de triage, donde los usuarios podrán interactuar con pacientes virtuales y evaluar signos vitales, así como realizar un cuestionario para determinar el nivel de triage del paciente. El objetivo de esta aplicación es ofrecer un entorno de aprendizaje más inmersivo y práctico que prepare a los estudiantes para situaciones clínicas reales.

1.3 Ámbito

Con respecto al problema identificado, se hace necesario desarrollar un software que complemente la aplicación de realidad aumentada y virtual, con el objetivo de brindar a los estudiantes de Técnico en Enfermería una experiencia de aprendizaje más robusta y efectiva. Este software permitirá a los usuarios acceder a un catálogo virtual de escenarios médicos, informándoles sobre los recursos y simulaciones disponibles. Además, los profesores podrán tener acceso en el portal web Educar , sobre el progreso de los estudiantes y cualquier área que requiera atención adicional.

El sistema también proporcionará análisis en tiempo real de la participación de los estudiantes en las simulaciones, lo que permitirá a los educadores ajustar la formación y los recursos según sea necesario. De esta manera, se busca que los estudiantes no solo se familiaricen con los procedimientos clínicos a través de la realidad aumentada, sino que también cuenten con herramientas para mejorar su aprendizaje y desempeño en situaciones reales.

1.4 Definiciones, acrónimos y abreviaciones

ACRÓNIMO	DESCRIPCIÓN
EDUCAR	Educación de realidad aumentada y virtual.
BD	Base de Datos
DAS	Documento de arquitectura de sistema
SW	Software
MER	Modelo entidad-relación
ISO	Organización Internacional de Normalización.
SQL Server	Microsoft SQL Server es un sistema de gestión de base de datos relacional.
POP-UP	Un pop-up, también conocido como ventana emergente o ventana pop-up.
TCP/IP	Protocolos de internet
SII	Servicio de Impuestos Internos
API	Application Programming Interface
Meta Quest 2	Dispositivo de realidad virtual desarrollado por Meta Platforms, utilizado para experiencias inmersivas de VR.
VR	Realidad Virtual
AR	Realidad Aumentada
Unity	Motor de desarrollo de videojuegos y aplicaciones en 2D y 3D.
Blender	Software de código abierto para modelado, animación y renderizado en 3D.
CRUD	CRUD es el acrónimo de Create (Crear), Read (Leer), Update (Actualizar) y Delete (Borrar)

1.5 Referencias

A continuación se listan las referencias a otros documentos:

1.5.1 Especificación de documentos

- La especificación de casos de uso
W Especificación_de_casos_de_uso (Terminado).docx .
- Acta de constitución
W PMOInformatica Plantilla Acta de Constitución del Proyecto.docx
- Carta Gantt o Cronograma de actividades ([Carta Gantt Educ-AR](#))
- Diccionario de datos (W Diccionario de datos.docx)

1.5.2 Resumen ejecutivo

EDUC-AR es una plataforma de simulación en realidad aumentada y virtual diseñada para estudiantes de Técnico en Enfermería en DUOC UC. El proyecto permite la práctica del triage clínico mediante escenarios interactivos que simulan la clasificación de pacientes en urgencias. Utilizando gafas Meta Quest 2, los estudiantes toman signos vitales y evalúan a los pacientes virtuales con distintos niveles de dificultad.

La plataforma además incluye un portal web para profesores, que permite monitorizar el progreso de los estudiantes en tiempo real.

El sistema EDU-AR está basado en un backend .NET Core, herramienta unity, blender para modelado 3d y con almacenamiento en Azure SQL Server, garantizando seguridad y accesibilidad.

El proyecto se presenta como una solución para mejorar la experiencia educativa en el área de salud, proporcionando una formación más práctica y realista para los futuros técnicos en enfermería.

1.6 Representación

La arquitectura del sistema EDUC-AR se basa en el enfoque del framework Modelo 4+1 y las recomendaciones del proceso unificado. Este enfoque asegura que todas las vistas relevantes del sistema estén adecuadamente representadas para cumplir con los objetivos funcionales y de calidad. En esta versión del documento, se incluyen las siguientes vistas:

1.6.1 Vista de Casos de Uso y Escenarios de Calidad

Los actores principales del sistema de Edu.car son el estudiante y profesor.

☐ **El estudiante:**

- Interactúa con la simulación de triage, donde realiza evaluaciones clínicas a través de realidad aumentada.
- Inicia sesión ingresando sus credenciales (correo y contraseña) para acceder al sistema de simulación.
- Selecciona el escenario de la simulación un nivel de dificultad (básico, medio o avanzado).
- En la simulación, el estudiante interactúa con el paciente virtual, tomando signos vitales como presión, pulso y temperatura.
- El estudiante accede a la ficha clínica del paciente y responde un cuestionario para clasificar el nivel de triage del paciente (C1 a C5).

☐ **El profesor:**

- El profesor ingresa sus credenciales (correo educacional y contraseña) en el portal web.
- Accede al portal web para revisar el progreso de los estudiantes y proporcionar retroalimentación en tiempo real.
- El profesor puede acceder a un panel con gráficos de avance de los estudiantes y, mediante un filtro, buscar a un estudiante por RUT.
- El sistema presenta el progreso del estudiante seleccionado, permitiendo al profesor descargar informes en formato PDF o Excel.

1.6.2 Vista de Metas y Restricciones

1. **Diagrama de Despliegue** : El sistema EDUC-AR está compuesto por varios nodos interconectados que aseguran la correcta ejecución y gestión de las simulaciones de triage en tiempo real.

- **Meta Quest 2 (Dispositivo del Estudiante):**

Los estudiantes acceden a la simulación de triage a través de los lentes Meta Quest 2, donde interactúan con pacientes virtuales y realizan procedimientos clínicos. Este dispositivo se conecta al servidor backend utilizando el protocolo HTTPS para garantizar la transmisión segura de datos en tiempo real.

- **Servidor Backend (Azure App Service):**

El servidor backend, alojado en Azure App Service, gestiona la lógica de la simulación y las interacciones del sistema. Es el encargado de procesar los datos enviados desde los lentes Meta Quest 2, como los resultados de las simulaciones y las respuestas a los cuestionarios de los estudiantes. Este servidor también está en comunicación con la base de datos y el portal web del profesor.

- **Base de Datos en Azure SQL:**

Toda la información relacionada con las simulaciones, incluidos los resultados de los estudiantes, las evaluaciones y los datos de progreso, se almacena en una Base de Datos en Azure SQL. El servidor backend consulta esta base de datos para almacenar y recuperar los datos necesarios para las simulaciones y los informes.

- **Portal Web para Profesores:**

Los profesores acceden al portal web para revisar el progreso de los estudiantes, descargar informes y proporcionar retroalimentación. Este portal está conectado al servidor backend, que extrae la información de la base de datos en Azure y la presenta en un formato visual a los profesores. El acceso al portal web también se realiza a través del protocolo HTTPS para garantizar la seguridad de los datos.

1.6.3.Vista Lógica

En la gestión de Educar podemos observar la perspectiva de los usuarios y el funcionamiento , también la descripción de la estructura del sistema Edu-car en el cual está representado por los siguientes diagramas de vista lógica.

- **Diagrama de Clases** : Se encuentra el modelo de entidad relación en cual se representan los siguientes módulos de clases representados.

1. **Clase Acción_Objeto** : Se utiliza para relacionar acciones específicas con los objetos de realidad aumentada (AR) en el sistema EDUC-AR.

Tiene los siguientes procesos principales como :

- ☐ **Agregar Acción:** Registra nuevas acciones que se pueden realizar con los objetos AR.
- ☐ **Consultar Acciones:** Permite obtener la lista de acciones disponibles para los objetos AR.
- ☐ **Modificar Acción:** Actualiza los detalles de las acciones existentes.

2. **Clase Acciones:** La **Clase Acciones** almacena las acciones realizadas por los estudiantes en el sistema durante las simulaciones.

Tiene los siguientes procesos principales como :

- ☐ **Registrar Acción:** Almacena las acciones clínicas o educativas realizadas por el estudiante.
- ☐ **Consultar Historial de Acciones:** Permite al profesor revisar las acciones realizadas por un estudiante en las simulaciones.
- ☐ **Modificar Registro de Acción:** Actualiza la información sobre las acciones realizadas.

3. **Clase Categorizacion_Triage** : Las categorizaciones de triage sirven para clasificar el nivel de urgencia de un paciente. En cual se encuentran las siguientes: C1 Atención inmediata, C2 Emergencia, C3 Urgencia, C4 Sin riesgo vital inmediato y C5 que es atención inmediata.

Tiene los siguientes procesos principales como :

- ☐ **Consultar Categorías:** Permite a los estudiantes acceder a las categorías existentes.
- ☐ **Modificar Categoría:** Actualiza la descripción o nivel de triage de una

categoría existente.

- ☐ **Eliminar Categoría:** Elimina una categoría de triage si ya no es relevante.

4. **Detalle_Evaluacion:** Almacena los detalles de las evaluaciones de los estudiantes , como preguntas, respuestas y si hubo errores.

Tiene los siguientes procesos principales como :

- ☐ **Consultar Evaluación:** Permite consultar los detalles de una evaluación específica.
- ☐ **Adicionar Detalle de Evaluación:** Registra los detalles de una nueva evaluación, incluyendo las respuestas y errores.
- ☐ **Modificar Detalle de Evaluación:** Actualiza los detalles de una evaluación en caso de que haya correcciones.

5. **Escenario_Virtual:** Define los escenarios virtuales utilizados en las simulaciones, con detalles como duración y nivel de dificultad.

Tiene los siguientes procesos principales como :

- ☐ **Agregar Escenario:** Registra un nuevo escenario virtual disponible para las simulaciones.
- ☐ **Consultar Escenarios:** Permite visualizar los escenarios de simulación disponibles.
- ☐ **Modificar Escenarios:** Actualiza los detalles de un escenario, como su nivel de dificultad o descripción.

6. **Ficha_Médica:** Contiene la información médica de los pacientes, incluyendo los detalles de las consultas, síntomas y tratamiento administrado.

Tiene los siguientes procesos principales como :

- ☐ **Agregar Ficha Médica:** Registra una nueva ficha médica para un paciente.
- ☐ **Consultar Escenarios:** Permite acceder a los detalles de la ficha médica de un paciente.
- ☐ **Modificar Escenarios:** Actualiza los detalles de un escenario, como su nivel de dificultad o descripción.

7. **Evaluación:** Las evaluaciones realizadas en el sistema Educ-ar por los estudiantes, serán almacenadas en la tabla.

Tiene los siguientes procesos principales como :

- ☐ **Agregar Evaluación :** Registra una nueva evaluación realizada por el estudiante.
- ☐ **Consultar Evaluación:** Permite visualizar las evaluaciones disponibles y sus resultados.
- ☐ **Modificar Evaluación :** Actualiza los detalles de una evaluación, como las respuestas o resultados..
- ☐ **Eliminar Evaluación:** Permite eliminar evaluaciones que estén

registradas por más de 5 años.

8. **Niveles:** Almacena los niveles de dificultad de la simulación, que incluyen los niveles básico, medio y avanzado. Estos niveles son utilizados para adaptar la simulación a la habilidad y progreso del estudiante, proporcionando un entorno de aprendizaje adecuado a su nivel de competencia.

Tiene los siguientes procesos principales como :

- ☐ **Agregar nivel :** Registra un nuevo nivel alcanzado por un estudiante en la simulación.
- ☐ **Consultar Niveles:** Permite acceder a los niveles alcanzados por los estudiantes.
- ☐ **Modificar Niveles :** Actualiza el nivel de un estudiante en caso de error o ajuste.

9. **Objetos_AR:** Todos los objetos de realidad aumentada (AR) utilizados en la simulación, se encuentran almacenados.

Tiene los siguientes procesos principales como :

- ☐ **Agregar Objeto AR :** Registra un nuevo objeto de realidad aumentada en el sistema.
- ☐ **Consultar Objetos AR:** Permite visualizar los objetos AR disponibles para la simulación.
- ☐ **Modificar Objeto AR :** Actualiza los detalles de un objeto AR, como su contenido o URL.

10. **Paciente:** Almacena la información de los pacientes registrados en el sistema, como su RUT, nombre, y nivel asignado.

Tiene los siguientes procesos principales como :

- ☐ **Agregar Paciente:** Registra un nuevo paciente en el sistema.
- ☐ **Consultar Pacientes:** Permite visualizar la información de los pacientes registrados.
- ☐ **Modificar Paciente:** Actualiza los detalles del paciente, como nombre o nivel.
- ☐ **Eliminar Paciente:** Elimina un paciente del sistema si ya no es necesario.

11. **Pregunta:** Almacena las preguntas utilizadas en las evaluaciones realizadas durante las simulaciones.

Tiene los siguientes procesos principales como :

- ☐ **Agregar Pregunta:** Registra una nueva pregunta en la base de datos.
- ☐ **Consultar Preguntas:** Permite acceder a las preguntas activas utilizadas en las evaluaciones.
- ☐ **Modificar Pregunta:** Actualiza el contenido de una pregunta existente.
- ☐ **Eliminar Pregunta:** Elimina una pregunta que ya no es relevante para las evaluaciones.

12. **Sintomas_Signos_Vitales:** Almacena los síntomas y signos vitales registrados de los pacientes durante las consultas simuladas.

Tiene los siguientes procesos principales como :

- ☐ **Adicionar Registro de Síntomas:** Registra nuevos síntomas y signos vitales de los pacientes.
- ☐ **Consultar Registros de Síntomas:** Permite acceder a los registros de signos vitales de los pacientes.
- ☐ **Modificar Registro de Síntomas:** Actualiza los síntomas y signos vitales de un paciente si hubo algún cambio.

13. **Tipo_Usuario:** Define los diferentes tipos de usuarios dentro del sistema, como estudiantes, profesores o administradores.

Tiene los siguientes procesos principales como :

- ☐ **Adicionar Tipo de Usuario:** Registra un nuevo tipo de usuario en el sistema.
- ☐ **Consultar Tipos de Usuario:** Permite visualizar los tipos de usuario disponibles.
- ☐ **Modificar Tipo de Usuario:** Actualiza la descripción de un tipo de usuario.

14. **Usuario:** Almacena la información de los usuarios registrados en el sistema, como sus credenciales y el tipo de usuario (profesor o estudiante).

Tiene los siguientes procesos principales como :

- ☐ **Adicionar Usuario:** Registra un nuevo usuario (profesor o estudiante) en el sistema.
 - ☐ **Consultar Usuarios:** Permite acceder a la información de los usuarios registrados.
 - ☐ **Modificar Usuario:** Actualiza los detalles del usuario, como nombre, correo o contraseña.
 - ☐ **Eliminar Usuario:** Elimina un usuario si ya no es necesario en el sistema.
-
- **Diagrama de Comunicación:** representan una combinación de información tomada desde el diagrama de clases, secuencia, y diagrama de casos de uso describiendo tanto la estructura estática como el comportamiento dinámico del sistema de Educ-car.
 - ❖ **Estudiante:** El estudiante inicia sesión en la Aplicación de Simulación usando sus credenciales.
 - ❖ Una vez autenticado, selecciona el escenario de simulación y comienza la práctica de triage.
 - ❖ Durante la simulación, el estudiante interactúa con el paciente virtual, toma signos vitales y completa el cuestionario de triage.
 - ❖ Al finalizar, el estudiante envía los resultados de la simulación al Servidor Backend.
 - ❖ **Servidor Backend (app y portal Web) :**El servidor recibe los resultados enviados por el estudiante y valida los datos.
 - ❖ El servidor almacena la información en la Base de Datos para mantener un registro de los resultados de las simulaciones.
 - ❖ También el servidor genera un resumen del desempeño del estudiante y lo envía al Portal Web del Profesor.
 - ❖ **BD en Azure:** La Base de Datos almacena todos los datos generados por los estudiantes.
 - ❖ Incluye los resultados de las simulaciones, respuestas a cuestionarios y progresos.

- ❖ Proporciona acceso al Servidor Backend para consultar la información de los estudiantes cuando sea necesario.
- ❖ **Profesor:** El profesor inicia sesión en el Portal Web utilizando sus credenciales.
- ❖ Una vez autenticado, el profesor puede acceder a los informes de progreso de los estudiantes.
- ❖ El profesor revisa los resultados de las simulaciones, puede filtrar por RUT y descargar informes en formatos PDF o Excel.
- ❖ También puede proporcionar retroalimentación a los estudiantes basada en el rendimiento observado en las simulaciones.

1. **Diagrama de Secuencia :** . El diagrama de secuencia describe la interacción al enfocarse en la secuencia de mensajes que se intercambian, junto con sus especificaciones de ocurrencia correspondientes en las líneas de vida.

- **Estudiante:**

El estudiante interactúa con la aplicación de simulación en Meta Quest 2 y el servidor backend en Unity. El flujo del proceso es el siguiente:

- ❖ **Solicitar acceso a la simulación:** El estudiante solicita acceder a la simulación desde su aplicación.
- ❖ **Validar las credenciales:** La app de simulación envía las credenciales al servidor backend para validación.
- ❖ **Confirmar credenciales con la base de datos:** El servidor backend consulta la base de datos para confirmar si las credenciales del estudiante son correctas.
- ❖ **Seleccionar escenario y nivel de simulación:** Una vez validado, el estudiante selecciona el escenario de simulación y el nivel de dificultad.
- ❖ **Cargar el escenario:** La aplicación carga el escenario seleccionado.
- ❖ **Iniciar simulación:** El estudiante inicia la simulación y comienza la interacción.
- ❖ **Interacción con el paciente virtual:** El estudiante realiza acciones en la simulación, interactuando con el paciente virtual.
- ❖ **Guardar las acciones de la simulación:** La aplicación guarda las acciones del estudiante durante la simulación.
- ❖ **Almacenar en la base de datos:** El servidor guarda los datos de las acciones en la base de datos.
- ❖ **Enviar resultados:** Al finalizar, el estudiante envía los resultados.
- ❖ **Guardar los resultados en la base de datos:** Los resultados de la simulación se almacenan en la base de datos.
- ❖ **Enviar y guardar en Azure BD:** El servidor envía los datos finales a la

base de datos en Azure.

- ❖ **Mostrar feedback de la simulación:** Finalmente, se le muestra al estudiante un resumen con feedback de su rendimiento en la simulación.

- **Profesor:**

El profesor accede al portal web EDUC-AR para revisar el progreso de los estudiantes y generar informes. El flujo es el siguiente:

- ❖ **Iniciar sesión:** El profesor inicia sesión en el portal web.
- ❖ **Validar las credenciales:** El portal web valida las credenciales del profesor consultando con el servidor backend.
- ❖ **Consultar credenciales:** El servidor consulta la base de datos para verificar las credenciales.
- ❖ **Consultar los resultados de un estudiante:** El profesor selecciona un estudiante para revisar su progreso en la simulación.
- ❖ **Confirmar credenciales:** El servidor confirma las credenciales del profesor en la base de datos.
- ❖ **Mostrar listado de estudiantes:** El portal web muestra el listado de estudiantes disponibles para consulta.
- ❖ **Ingresar RUT del estudiante:** El profesor ingresa el RUT del estudiante para filtrar los resultados, formato (10234567-8).
- ❖ **Presionar botón filtrar:** El profesor presiona el botón de filtrar para buscar el estudiante específico.
- ❖ **Consultar en base de datos:** El portal web consulta en la base de datos del servidor la información del estudiante.
- ❖ **Consultar RUT estudiante:** El servidor consulta en la base de datos la información relacionada con el estudiante.
- ❖ **Enviar los datos:** El servidor envía los datos del estudiante al portal web.
- ❖ **Mostrar información de la base de datos:** El portal web muestra la información recuperada del estudiante.
- ❖ **Generar informe:** El profesor genera un informe basado en los resultados obtenidos del estudiante.
- ❖ **Seleccionar formato PDF o EXCEL :** El profesor elige generar el informe en formato PDF o Excel.
- ❖ **Imprimir informe según formato seleccionado:** Finalmente, el informe en formato PDF o Excel se genera e imprime.

1.6.4 Vista de Procesos

1. Diagrama de Actividad: es un diagrama que muestra el flujo de control o el flujo de objetos con énfasis en la secuencia y las condiciones del flujo. Las acciones coordinadas por los modelos de actividad se pueden iniciar porque otras acciones terminan de ejecutarse.

- Este diagrama de actividad describe el flujo de interacción entre el profesor (a través del portal web) y los estudiantes (a través de la aplicación de realidad aumentada). Los estudiantes utilizan la aplicación para realizar simulaciones interactivas de triage, mientras que el profesor supervisa el progreso a través del portal web. Los actores principales son el Profesor y los Estudiantes.

1. **Profesor:** Utiliza el portal web para ver el progreso de los estudiantes.

- ❖ **Inicio de Sesión:** El profesor accede al portal web de EDUC-AR e ingresa sus credenciales.
- ❖ **Ver Progreso de Estudiantes:** Desde el menú principal, selecciona la opción para ver el progreso de los estudiantes.
- ❖ **Consulta de Base de Datos:** El sistema consulta la base de datos para obtener la lista de estudiantes y su progreso en las simulaciones de triage.
- ❖ **Visualización de Resultados:** El profesor ve los resultados generales de los estudiantes.
- ❖ **Revisión de Detalles:** Selecciona un estudiante para ver un informe detallado de sus decisiones y rendimiento en la simulación.
- ❖ **Cierre de Sesión:** El profesor cierra la sesión y sale del sistema.

2. **Estudiantes:** Utilizan la app de realidad aumentada para interactuar con las simulaciones.

- ❖ **Inicio de Sesión en la App:** El estudiante accede a la aplicación de simulación en los lentes de realidad aumentada Meta Quest 2.

- ❖ **Selección de Escenario:** El estudiante elige un escenario de simulación (nivel de dificultad) y lo inicia.
- ❖ **Interacción en Simulación:** Realiza las acciones de triage, como tomar signos vitales, evaluar al paciente, y tomar decisiones médicas.
- ❖ **Finalización de Simulación:** Al completar la simulación, se genera un informe de desempeño.
- ❖ **Almacenamiento de Resultados:** Los resultados de la simulación se guardan en la base de datos del sistema.
- ❖ **Enviar Evaluación o finalizar sesión :** El estudiante puede revisar un resumen de sus aciertos y errores al finalizar.

2. Vista de Implementación

2.1 Diagrama de Componentes:

La estructura del proyecto **EducAR** tiene 5 componentes clave:

1. **Web Server:** A través de una página de administración hecha en **C#**, gestiona el sistema administrativo del proyecto.
2. **SQL Server:** Aloja la base de datos en **Azure**, conectada al Web Server y a la API para gestionar los datos.
3. **API:** Desarrollada en **C#**, permite la comunicación entre la base de datos y otros componentes, como la aplicación de realidad aumentada.
4. **Firebase:** Proporciona autenticación de usuarios, necesaria para acceder a la aplicación AR y otros servicios del proyecto.
5. **Aplicación AR:** Creada en **C#**, **Unity**, y **Blender**, es la aplicación interactiva que utiliza los servicios del API y Firebase para brindar la experiencia de realidad aumentada.

Todos los componentes están integrados para garantizar una autenticación segura y un flujo constante de información entre la aplicación, el API y la base de datos.

2.1 Diagrama de Paquetes:

El diagrama de paquetes desglosa los diferentes módulos del sistema en paquetes funcionales.

1. **Paquete de Interfaz Web de Administración:**
 - Es el encargado de la interfaz de administración del sistema. Este paquete interactúa con el paquete de API para hacer solicitudes y obtener datos que luego se muestran o gestionan en la interfaz de administración.
2. **Paquete de API:** Este paquete se encarga de procesar las solicitudes de datos y gestionar la lógica del negocio. Está conectado al Paquete de Base de Datos para recuperar o almacenar información y a Firebase para realizar autenticaciones. También se conecta con la Aplicación AR para gestionar la lógica de la realidad aumentada y con el Paquete de Interfaz Web de Administración para facilitar la

administración.

3. **Paquete de Base de Datos:** Este paquete contiene la lógica para interactuar con la base de datos, almacenando y recuperando información según las solicitudes del Paquete de API. Está ubicado de manera central, ya que tanto la API como la Aplicación AR interactúan con él.
4. **Firebase:** Está separado como un componente de autenticación. Tanto el Paquete de API como la Aplicación AR interactúan con Firebase para gestionar el acceso de los usuarios.
5. **Paquete de Aplicación AR:** Se encarga de la funcionalidad de la aplicación de realidad aumentada. Este paquete interactúa con el Paquete de API para acceder a los datos y autenticarse con Firebase. Además, interactúa con el Paquete de Base de Datos para almacenar y recuperar datos relacionados con la experiencia AR.

3 Metas y Restricciones de la Arquitectura

A continuación se revisan las metas y restricciones de la arquitectura.

3.1 Metas de la arquitectura

- **Usabilidad**

La arquitectura propuesta busca ofrecer una interfaz fácil de usar tanto para los estudiantes como para los administradores del sistema EDUC-AR. Los estudiantes deben poder acceder a simulaciones de realidad aumentada (AR) y contenidos educativos desde navegadores y dispositivos móviles, mientras que los administradores podrán monitorear el rendimiento y el

progreso de los estudiantes mediante una interfaz web intuitiva.

- **Confiabilidad y Seguridad**

La seguridad es un pilar clave en la arquitectura. Firebase Authentication garantiza la autenticación segura de los usuarios y protege los datos sensibles a través de encriptación y controles de acceso. Además, se prioriza la alta disponibilidad y el monitoreo constante para mitigar cualquier amenaza potencial. La confiabilidad se logra a través de una infraestructura robusta basada en Azure que garantiza tiempo de actividad continuo.

- **Rendimiento**

El rendimiento es crítico, especialmente en simulaciones en tiempo real con AR. La API de .NET Core debe ser eficiente y capaz de manejar múltiples solicitudes concurrentes, optimizando las consultas a SQL Server para asegurar una rápida respuesta a las solicitudes del sistema. Además, se implementarán cachés y balanceo de carga para mejorar el rendimiento general del sistema.

- **Soporte**

El sistema está diseñado para ser fácilmente mantenido y actualizado. Las tecnologías elegidas, como .NET Core, SQL Server y Firebase, proporcionan un soporte a largo plazo y facilitan la integración de nuevas funcionalidades sin necesidad de reestructurar la arquitectura básica. El uso de Azure permite administrar y actualizar los servicios sin interrupciones significativas.

Requerimientos Físicos

El sistema se aloja en la nube, específicamente en Azure, eliminando la necesidad de hardware físico adicional. Los servicios de infraestructura como base de datos, API y autenticación serán escalables, optimizando los recursos para ofrecer alta disponibilidad y respuesta rápida a los usuarios. Azure permite el aprovisionamiento de recursos bajo demanda, optimizando el costo según las necesidades del proyecto.

3.2 Restricciones de la Arquitectura

- **Tecnológicas**

La arquitectura está restringida a las tecnologías definidas en el proyecto: **.NET Core, SQL Server, Firebase Authentication y Unity**. Cualquier integración adicional debe estar alineada con estas plataformas y su compatibilidad.

- **Presupuestarias**

La utilización de servicios en la nube implica un control sobre los costos, especialmente en **Azure y Firebase**. Es importante que el uso de recursos sea eficiente para no exceder el presupuesto asignado.

- **Tiempo de desarrollo**

La arquitectura debe estar alineada con los plazos del proyecto. Las decisiones de diseño deben equilibrar la complejidad técnica con la necesidad de entregar el sistema en un plazo razonable, lo que limita la inclusión de soluciones altamente personalizadas.

3.3 Otros antecedentes y consideraciones

- **Evolutividad**

La arquitectura debe ser lo suficientemente flexible para permitir la incorporación de nuevas tecnologías, como otros dispositivos de realidad aumentada o futuras mejoras en las aplicaciones web y AR.

- **Escalabilidad**

El sistema debe ser capaz de escalar a medida que aumente el número de usuarios, tanto en la capacidad de procesamiento de la API como en el almacenamiento de datos en **SQL Server**.

- **Compatibilidad con Dispositivos**


La arquitectura está diseñada para ser compatible con los dispositivos de realidad aumentada actuales, como **Meta Quest 2**, pero debe tener la flexibilidad de integrarse con otros dispositivos que puedan ser añadidos en el futuro.

4 Vista de Casos de Uso y Escenarios de Calidad

Esta sección describe en detalle el conjunto de escenarios que obtuvieron la mayor prioridad en el análisis realizado a los procesos en la farmacia. Para esto se presenta y describe el diagrama de casos de uso y los casos de uso prioritarios en torno al proceso , así como los escenarios en que uno o más atributos de calidad se ven involucrados de manera significativa.

4.3 Modelo de Casos de Uso

El modelo de casos de uso puede ser encontrado en el documento

 Especificación_de_casos_de_uso (Terminado).docx

4.2 Especificación de Casos de Uso Relevantes

Los casos de uso considerados los más relevantes para el desarrollo de la arquitectura fueron determinados. Los criterios usados para dicha determinación fueron:

- Se identificaron aquellos casos de uso que son fundamentales para la interacción de los actores principales (Estudiante y Profesor) con el sistema. Esto incluye acciones críticas que impactan directamente la experiencia de aprendizaje y la gestión de resultados.
- Se consideraron los casos de uso que tienen un impacto significativo en los objetivos generales del proyecto, como mejorar la formación práctica de los estudiantes en situaciones de urgencia y facilitar la evaluación del progreso por parte de los profesores.
- Se tomó en cuenta cómo los diferentes casos de uso se interrelacionan. Los casos que forman parte de flujos de trabajo más amplios o que dependen de otros casos de uso fueron considerados relevantes para asegurar una integración adecuada en la arquitectura del sistema.

A continuación se listan los casos de uso relevantes, los cuales pueden ser encontrados con su especificación detallada en el documento “Casos de Uso”.

Código	Nombre	Actores	Prioridad
CU-001	Iniciar Sesión en la Aplicación	Estudiante	Alta
CU-002	Seleccionar Escenario de Simulación	Estudiante	alta
CU-003	Realizar Procedimiento Clínico de Triage	Estudiante	alta
CU-101	Iniciar Sesión en el Portal de Profesor	Profesor	alta
CU-102 -103	Visualización de Progreso y Descarga de Informe	Profesor	alta

4.3 Especificación de los Escenarios de Calidad Relevantes

Después de un análisis en conjunto con los stakeholders, los escenarios de calidad se expresan a continuación:

ID: QS1

Nombre: Desempeño: Tiempo de respuesta en la simulación de triage.

Síntesis: Tiempos de respuesta en la simulación de triage que no afecten el proceso de aprendizaje del estudiante.

Entorno: Proceso normal de operación del sistema durante las simulaciones.

Cambio en el entorno: Ejecución de simulaciones múltiples en paralelo.

Comportamiento esperado: El sistema debe procesar las simulaciones sin retrasos perceptibles.

Medida: Cada simulación debe comenzar en menos de 5 segundos tras la selección del escenario.

Prioridad Arquitectónica: Alta

Aplicación: Global

ID: QS2

Nombre: Tolerancia a fallos: Recuperación del sistema ante un fallo.

Sinopsis: El sistema debe poder restablecerse rápidamente ante fallos inesperados.

Entorno: El sistema está trabajando en su carga normal durante las simulaciones.

Cambio en el entorno: El servidor experimenta un fallo.

Comportamiento esperado: El sistema debe restablecerse y permitir que los estudiantes continúen con sus simulaciones.

Medida: El tiempo de recuperación no debe exceder los 15 minutos.

Prioridad Arquitectónica: Alta

Aplicación: Global

ID: QS3

Nombre: Usabilidad: Facilidad de navegación en el Portal del Profesor.

Sinopsis: El portal debe ser intuitivo y fácil de usar para los profesores al acceder a la información de los estudiantes.

Entorno: Uso normal del portal por parte de profesores.

Cambio en el entorno: Acceso desde diferentes dispositivos (PC, tabletas).

Comportamiento esperado: Los profesores deben poder acceder a sus funciones principales sin confusión.

Medida: Al menos el 90% de los profesores deben completar tareas clave sin asistencia.

Prioridad Arquitectónica: Media

Aplicación: Específica para el Portal del Profesor.

ID: QS4

Nombre: Seguridad: Protección de datos de los estudiantes.

Sinopsis: Los datos personales y resultados de los estudiantes deben estar protegidos contra accesos no autorizados.

Entorno: Proceso normal de operación del sistema.

Cambio en el entorno: Intento de acceso no autorizado a la base de datos.

Comportamiento esperado: El sistema debe bloquear el acceso y registrar el intento.

Medida: Todos los accesos no autorizados deben ser registrados y notificados al equipo técnico en menos de 5 minutos.

Prioridad Arquitectónica: Alta

Aplicación: Global

5. Vista Lógica

A continuación se presenta una vista lógica de la aplicación expresado en dos diagramas, uno de los diagramas permite capturar los elementos que integran la estructura interna con sus operaciones del sistema así como sus relaciones y otra vista se representan MER que emplea un conjunto definido de símbolos y líneas de conexión para representar la interconexión de entidades, relaciones ,sus atributos y Clave foránea.

5.1 Parte Estructural

En el siguiente diagrama de Módulos se observa que el módulo principal.

Ilustración 1: Diagrama de Clases

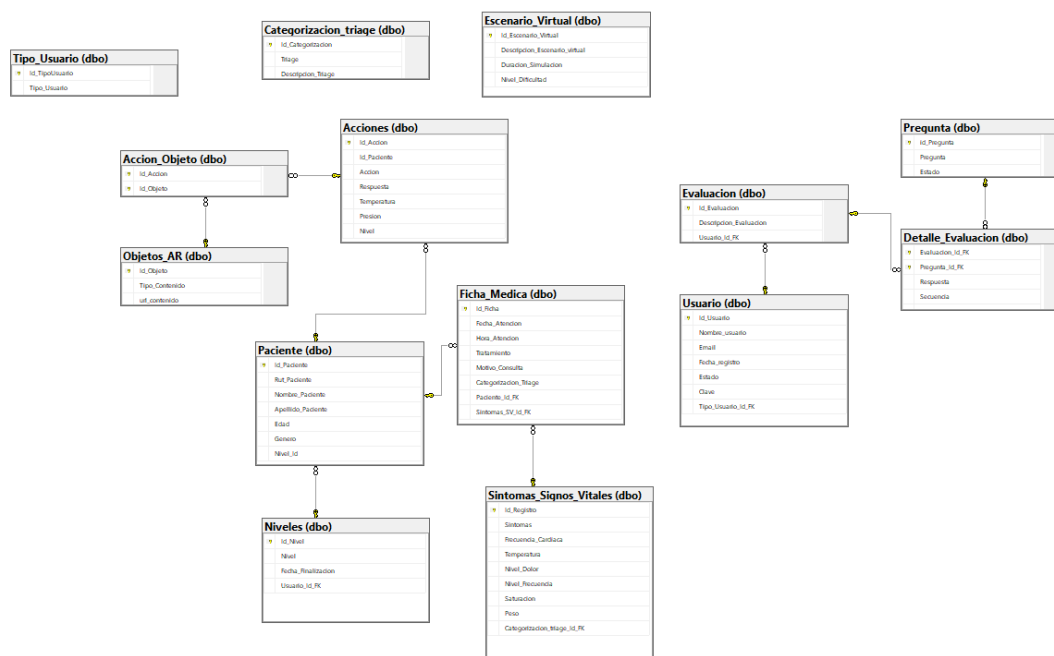


Ilustración 2: Diagrama de Clases Datamodel

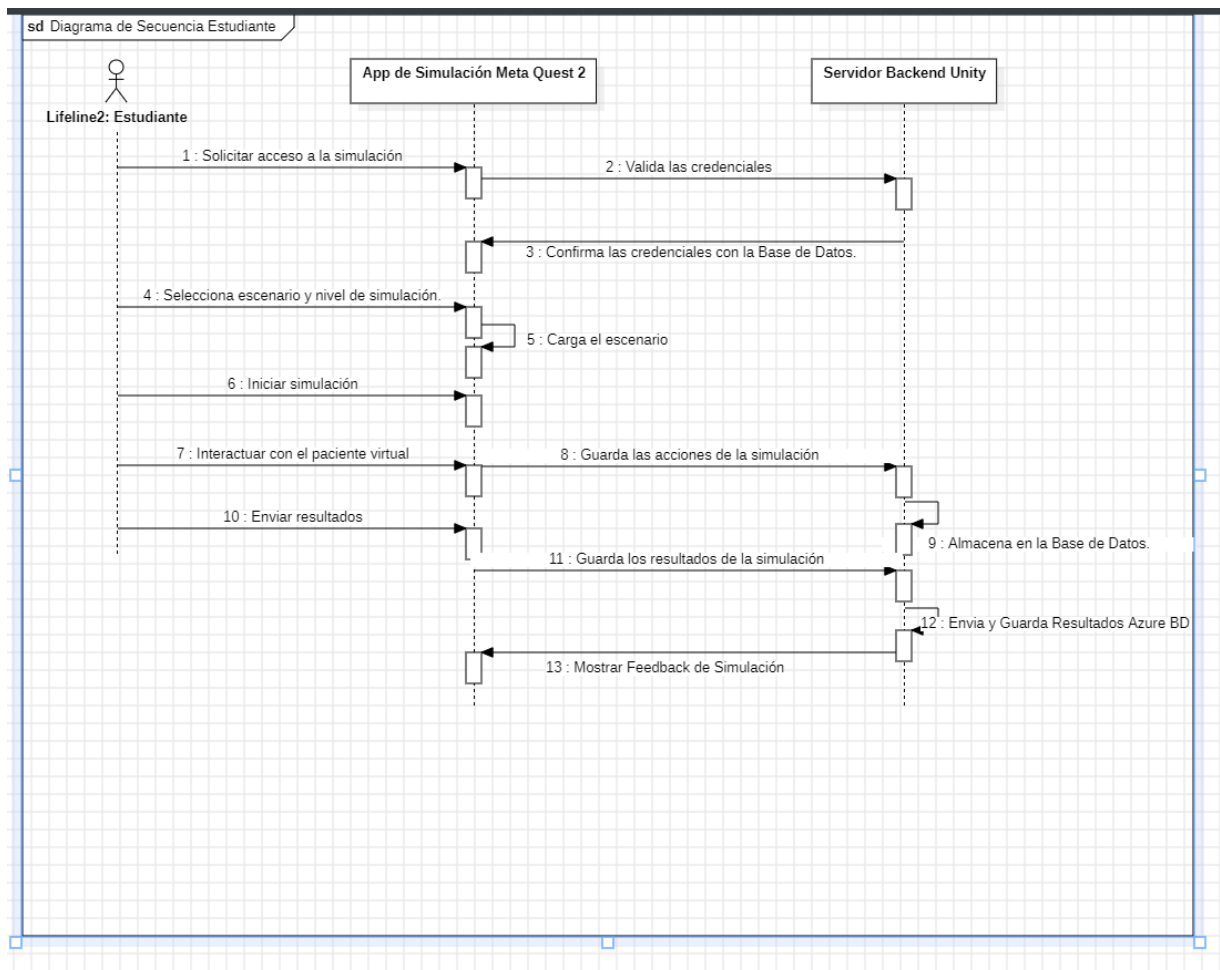


5.2 Parte Dinámica

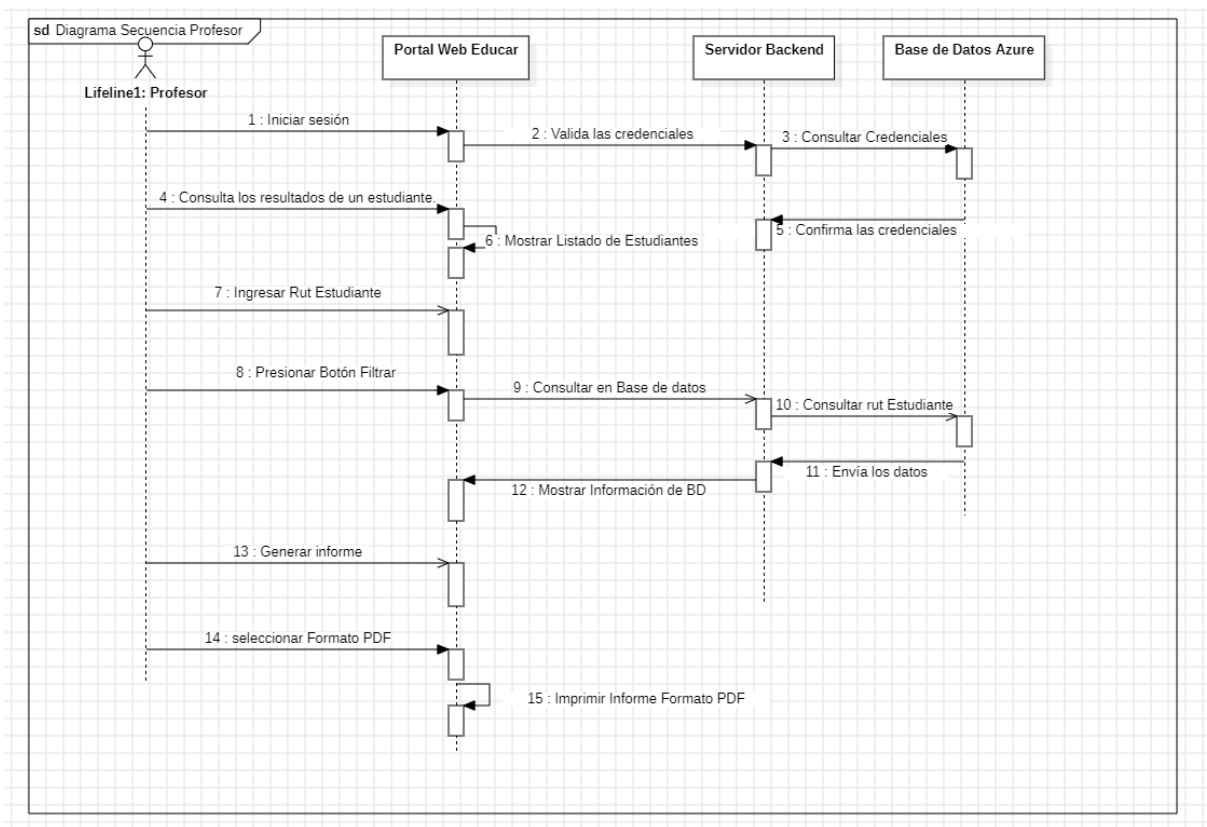
En este apartado podemos observar los diagramas de secuencia y comunicación separados por actor

Ilustración 2: Diagrama de Componentes y Conectores

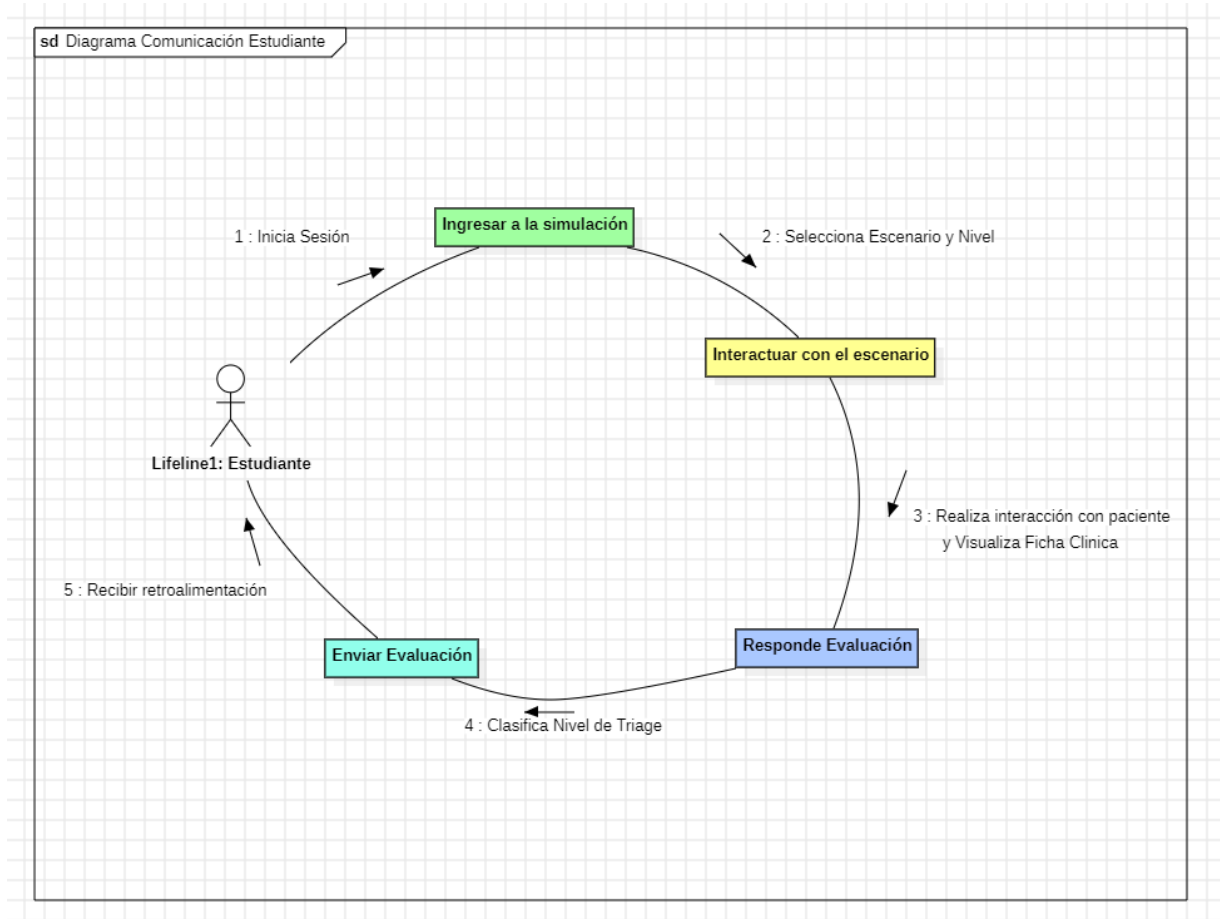
2.1: Secuencia Estudiante



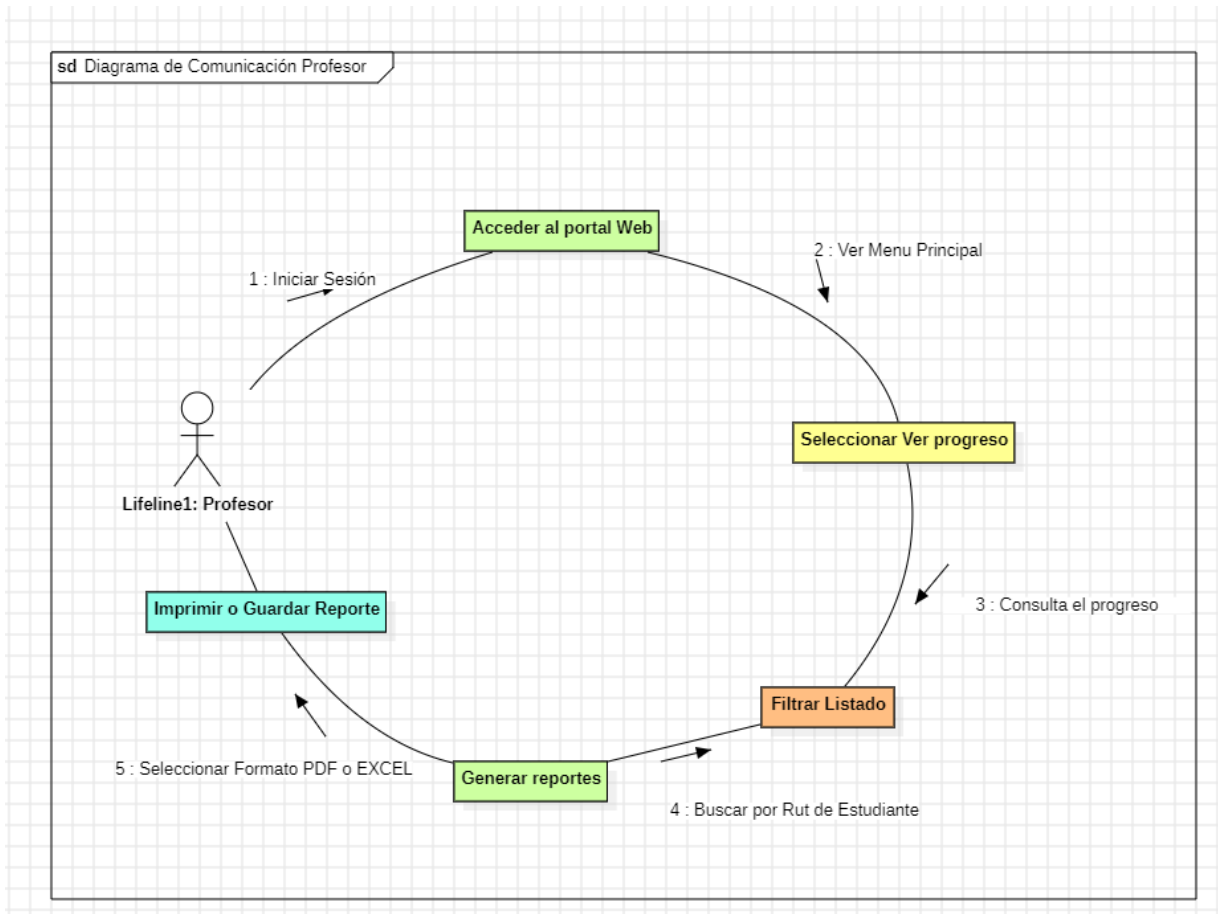
2.2: Secuencia Profesor



2.3 Diagrama de Comunicación Estudiante



2.4 Diagrama de Comunicación Profesor



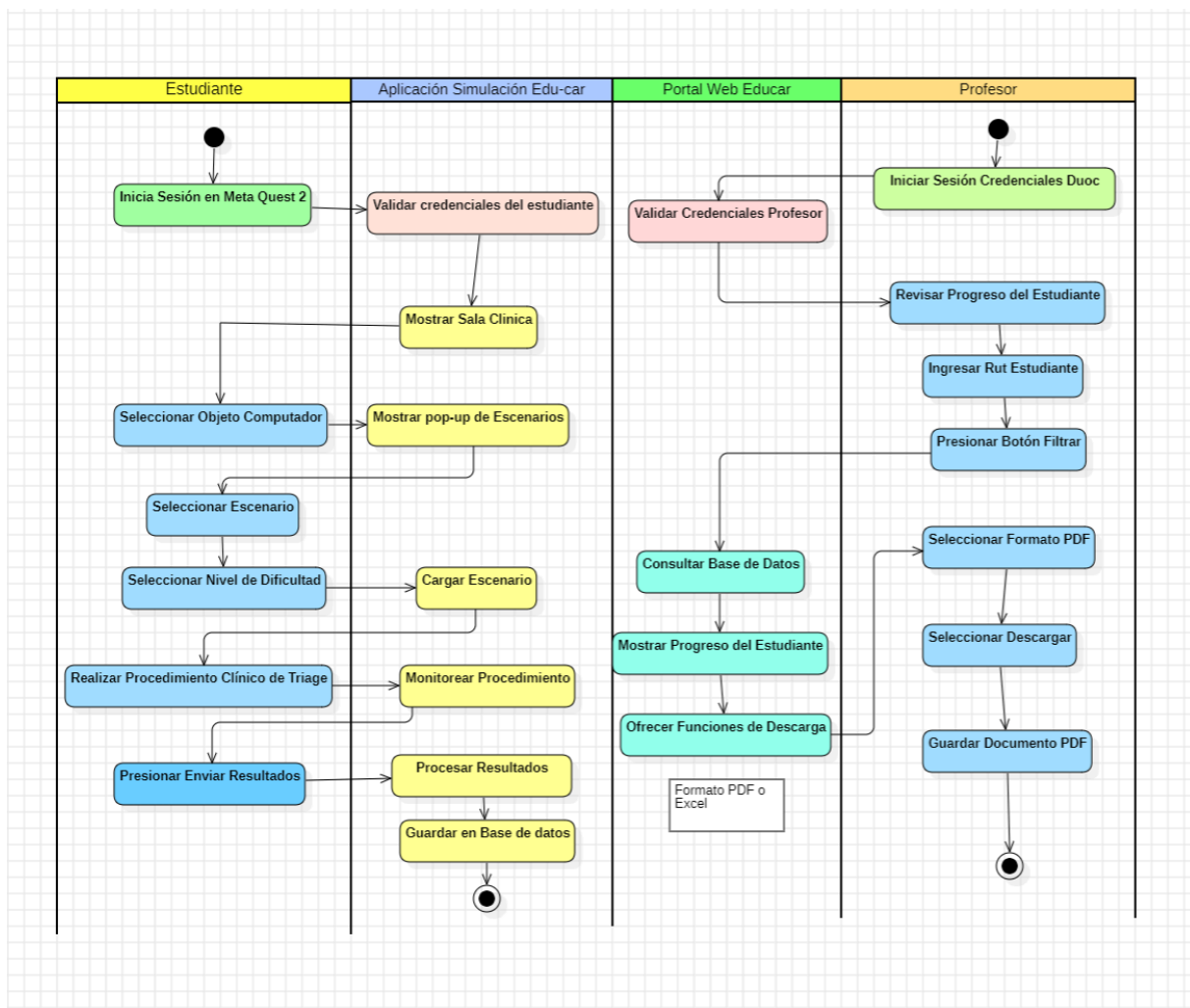
6. Vista de Procesos

La Vista de Procesos describe cómo los actores y componentes del sistema interactúan entre sí en el flujo de trabajo general de EDUC-AR. Para este caso, tanto el Estudiante como el Profesor realizan acciones importantes, pero la mayoría de las interacciones comienzan con el estudiante, ya que es quien realiza la simulación de triage. Luego, el profesor revisa los resultados a través del Portal Web.

- **Actores y Componentes:**

1. **Estudiante:** Interactúa con la Aplicación de Simulación (en los lentes Meta Quest 2), realiza la simulación de triage y envía los resultados.
2. **Profesor:** Accede al Portal Web para revisar el desempeño de los estudiantes y descargar los informes.
3. **Aplicación de Simulación EDU-CAR:** Sistema donde el estudiante ejecuta la simulación, toma decisiones clínicas, y genera resultados.
4. **Portal Web EDU-CAR:** Sistema donde el profesor accede a los informes de progreso de los estudiantes.

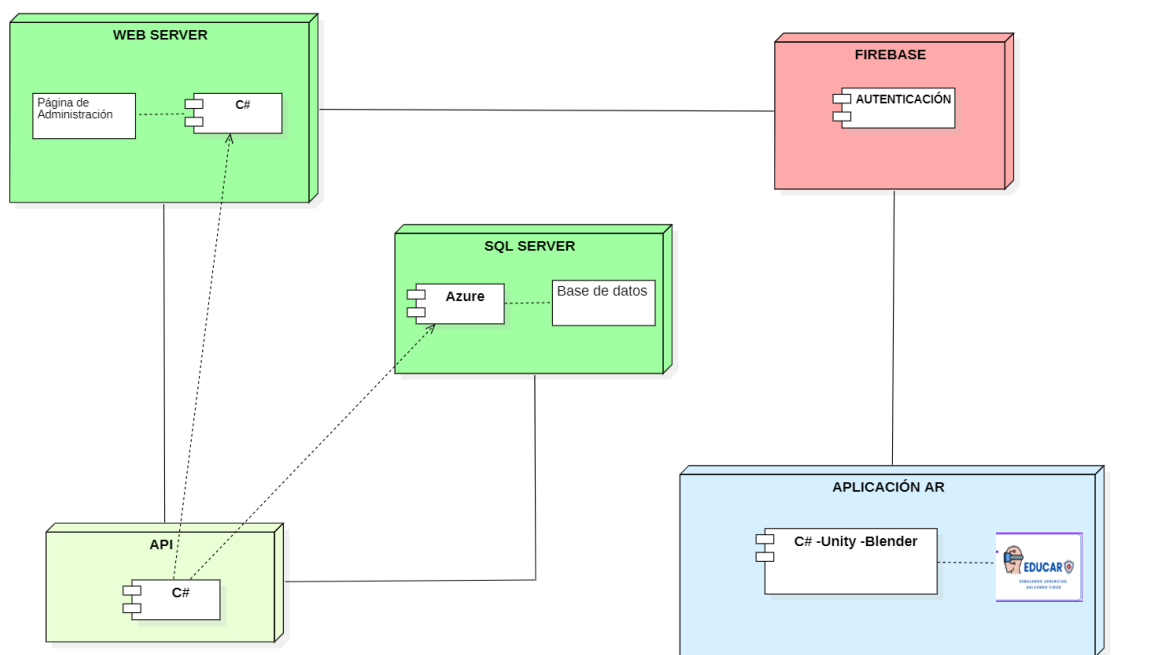
Ilustración 3: Diagrama de Actividad



7 Vista de Implementación

En esta vista se apreciará el modelo de vista física en cual se representa las máquinas físicas se representan como nodos. El nodo es un elemento donde se ejecutan los componentes. Los componentes embebidos son aplicaciones, librerías y motor de base de datos. Entre los nodos y componentes se comunican a través de relaciones lo que indica el tipo de conexión entre ellos.

Ilustración 4: Diagrama de Despliegue



8. Vista de Despliegue

En esta vista se despliegan los nodos que participan con el sistema. Los nodos principales son los nodos Servidor de Integración. Características a continuación:

El diagrama de arquitectura de software ilustra la interacción entre cinco componentes principales:

- Paquete de Interfaz Web de Administración:** Representado por un rectángulo negro en la parte superior izquierda.
- Paquete de API:** Representado por un rectángulo negro en la parte superior central.
- Paquete de Base de Datos:** Representado por un rectángulo negro en la parte central.
- Paquete de Aplicación AR:** Representado por un rectángulo negro en la parte inferior izquierda.
- Firebase:** Representado por un rectángulo negro en la parte superior derecha.

Las conexiones entre los paquetes se indican mediante líneas:

- Líneas sólidas:**
 - Del Paquete de Interfaz Web de Administración al Paquete de API.
 - Del Paquete de API al Paquete de Base de Datos.
 - Del Paquete de Base de Datos al Paquete de Aplicación AR.
 - Del Paquete de Base de Datos a Firebase.
- Líneas punteadas:**
 - Del Paquete de Interfaz Web de Administración al Paquete de Base de Datos.
 - Del Paquete de Aplicación AR al Paquete de Base de Datos.
 - Del Paquete de API a Firebase.

9. Decisiones de Diseño y Selección de Alternativas

Durante la fase inicial del proyecto EDUC-AR, se aplicaron diversas técnicas y métodos para evaluar y seleccionar la mejor alternativa, utilizando criterios de valoración que aseguren que cada opción cumpla con los requisitos establecidos en el diseño.

El objetivo de este proyecto, que simula el triage en realidad aumentada para estudiantes de Técnico en Enfermería, es garantizar el cumplimiento de los requisitos tanto a nivel teórico como práctico, basándose en el diseño arquitectónico de modelos 4+1. Estos modelos representan el comportamiento de los escenarios de simulación priorizados durante el análisis del sistema.

Decisiones tomadas en el desarrollo del proyecto:

1. **Metodología Tradicional:** Se adoptará un enfoque de gestión tradicional para asegurar una planificación estructurada y un control riguroso de las fases del proyecto, garantizando que se cumplan todos los plazos y requisitos.
2. **Plazo del Proyecto:** El proyecto tendrá una duración de **5 meses**, de acuerdo con la planificación general.
3. **Arquitectura CMS Headless:** Se utilizará una **arquitectura Headless** que permite desacoplar el backend del frontend, brindando flexibilidad para que múltiples dispositivos (web, móviles, AR) consuman el mismo contenido a través de una API. Esto facilita la escalabilidad, ya que cada canal puede tener su propia interfaz personalizada sin afectar la lógica del backend. Además, garantiza una mejor gestión de contenido centralizada y asegura que podamos integrar nuevas tecnologías o plataformas sin reestructurar todo el sistema.
4. **Funcionalidades CRUD en Azure:** Se implementarán las funcionalidades de los métodos CRUD en la sección de **Azure**, permitiendo la gestión eficaz de datos.
5. **Acceso a través de Internet:** Las funcionalidades del sistema serán accesibles mediante un navegador, facilitando la visualización y uso por parte de los estudiantes y docentes.
6. **Servidor de Base de Datos:** Se utilizará **Microsoft SQL y Azure** como servidor de base de datos.
7. **Normas de Calidad y Seguridad:** Se garantizará el cumplimiento de las normas **ISO/IEC 25000:2005**, así como de las normas de seguridad para el desarrollo de software, implementando la **ISO/IEC 15408**.
8. **Documentación del Proyecto:** Se generará documentación que permita un mejor entendimiento del proyecto y su aprobación por parte del cliente, que incluirá:
 - Planilla de requerimientos.
 - Modelo **4+1**.
 - Documento de Aceptación de Software (DAS).

10. Análisis de Reutilización

La reutilización nos permite simplificar el trabajo usando componentes ya construidos, optimizando los tiempos y enfocando los esfuerzos en lo nuevo.

Los componentes reutilizados son trozos de código encapsulados que devuelven una serie de utilidades (por ejemplo, una API).

- **Reutilización de Modelos en Blender**

En el desarrollo de EDUC-AR, se emplearán modelos de objetos de realidad aumentada (AR) reutilizados, creados previamente en Blender. Esto nos permitirá no solo ahorrar tiempo en la fase de diseño, sino también asegurar la calidad y coherencia visual de los objetos en las simulaciones.

- **Reutilización de Código**

Además de los modelos, en la creación de las clases de Azure, reutilizamos el código de los métodos de crear, obtener, actualizar y eliminar, lo que reduce el tiempo de desarrollo y mejora la productividad grupal.

Los componentes ya probados y depurados anteriormente a través de pruebas unitarias y de integración se verificarán para asegurarnos de que los métodos funcionen correctamente tras la reutilización del código. Debido a los riesgos asociados con la modificación de estos componentes, el plan del proyecto reflejará la utilización de estos elementos, garantizando así una mayor eficiencia y efectividad en el desarrollo del sistema.