



**Puesta en valor de un  
simulador de entrenamiento  
mediante la incorporación  
de experiencia inmersiva  
y analíticas de aprendizaje**

---

**Autor:** Franco Lanzillotta  
**Directora:** Dra. Stella Maris Massa  
**Codirector:** Ing. Adolfo Tomás Spinelli

– Octubre 2020 –



RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-  
NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).



**Puesta en valor de un  
simulador de entrenamiento  
mediante la incorporación  
de experiencia inmersiva  
y analíticas de aprendizaje**

---

**Autor:** Franco Lanzillotta  
**Directora:** Dra. Stella Maris Massa  
**Codirector:** Ing. Adolfo Tomás Spinelli

– Octubre 2020 –

## **Agradecimientos**

A mis padres, por apoyarme incondicionalmente, por haberme brindado su soporte durante toda mi carrera y haberme permitido llegar hasta acá.

A mi hermana Rocío, por aguantarme con mis virtudes y defectos y siempre estar presente para mí.

A mi novia Wendy, por acompañarme en cada paso que doy y siempre ayudarme a progresar y salir adelante.

A mis amigos, por alegrarme siempre con su compañía y apoyarme en los buenos y malos momentos.

A mis compañeros, por haber transitado conmigo este camino y haberlo hecho mucho más ameno y entretenido.

A mis directores, Stella y Adolfo, por guiarme y acompañarme durante todo el desarrollo de este trabajo.

A la Facultad de Ingeniería, por darme la oportunidad de haber estudiado esta carrera y haberme brindado todos los recursos.

A Julio, Nicolás y Roberto de la Fuerza Aérea Argentina, por haberme recibido siempre con los brazos abiertos y haber hecho todo lo posible para llegar a concretar este proyecto.

## Resumen

Los simuladores de entrenamiento son utilizados para generar entornos simulados que imitan la realidad en mayor o menor medida, a través de los cuales los usuarios pueden entrenar distintas habilidades y adquirir nuevos conocimientos. De esta manera, posibilita capacitar sin incurrir en costos operativos y sin los riesgos inherentes del escenario real. Sin embargo, la evaluación del aprendizaje y de las habilidades adquiridas durante la simulación es una tarea muy compleja. Por lo tanto, hacen falta métodos que permitan recopilar y analizar los datos resultantes de las interacciones de los aprendices, para poder así medir su desempeño y producir evidencias válidas del aprendizaje.

En este marco, se ha propuesto llevar a cabo la puesta en valor de un simulador de entrenamiento para el uso de artillería antiaérea, incorporando mejoras en la simulación para aumentar la experiencia inmersiva, conjuntamente con la medición de datos y presentación de informes. Este sistema inspecciona cómo interactúa cada aprendiz con el simulador, almacenando información detallada y generando informes sobre las interacciones y los cambios en el estado interno de la simulación para un análisis posterior. De esta manera, permite al instructor evaluar objetivamente el desempeño del aprendiz, además de poder medir su progreso a lo largo de las sesiones de entrenamiento.

**Palabras clave:** *Juegos Serios, Simuladores, Analíticas de Aprendizaje, Interacción Persona-Computadora.*

## Abstract

Training simulators are used to generate simulated environments that imitate reality to a greater or lesser extent, through which users can train different skills and acquire new knowledge. In this way, it enables training without incurring operating costs and without the inherent risks of the real scenario. However, the evaluation of learning and skills acquired during simulation is a very complex task. Therefore, methods are needed to collect and analyze the data resulting from the interactions of the learners, in order to measure their performance and produce valid evidence of learning.

In this context, it has been proposed to carry out the enhancement of a training simulator for the use of anti-aircraft artillery, incorporating improvements in the simulation to increase the immersive experience, together with the measurement of data and reporting. This system inspects how each learner interacts with the simulator, storing detailed information and generating reports on the interactions and changes in the internal state of the simulation for further analysis. In this way, it allows the instructor to objectively assess the learner's performance, in addition to being able to measure their progress throughout the training sessions.

**Keywords:** *Serious Games, Simulators, Learning Analytics, Human-Computer Interaction.*

## Índice

<b>Capítulo 1: Introducción.....</b>	<b>7</b>
1.1 Objetivos.....	8
1.2 Publicaciones vinculadas.....	9
<b>Capítulo 2: Marco teórico.....</b>	<b>10</b>
2.1 Juegos Serios.....	10
2.2 Simuladores de Entrenamiento.....	11
2.3 Analíticas de Videojuegos.....	12
2.3.1 Analíticas de Aprendizaje.....	13
2.3.2 Analíticas de Aprendizaje en Juegos Serios.....	14
<b>Capítulo 3: Estado del arte.....</b>	<b>16</b>
3.1 Metodología.....	16
3.2 Planificación de la revisión.....	17
3.2.1 Identificación de las necesidades que motivan la revisión.....	17
3.2.2 Definición de un protocolo de revisión.....	17
3.3 Resultados de la revisión.....	18
3.4 Discusión y conclusiones del estudio.....	22
<b>Capítulo 4: Metodología.....</b>	<b>24</b>
4.1 Modelo de Proceso de Desarrollo para Juegos Serios.....	24
4.2 Marco de trabajo para la vinculación de Analíticas de Aprendizaje en el diseño de Juegos Serios.....	28
<b>Capítulo 5: Trabajo experimental.....</b>	<b>31</b>
5.1 Problema a resolver.....	31
5.2 Dominio del problema.....	34
5.3 Elicitación.....	36
5.4 Especificación.....	36
5.4.1 Definición de los requerimientos.....	36
5.4.2 Selección de Analíticas de Aprendizaje.....	38
5.4.3 Parametrización de la simulación.....	39
5.5 Diseño.....	41
5.5.1 Arquitectura del sistema.....	41
5.5.2 Analíticas de Aprendizaje.....	43

---

5.6 Producción.....	44
5.6.1 Selección de motores de videojuegos.....	44
5.6.2 Simulador.....	47
5.6.2.1 Simulador.....	47
5.6.2.2 Gestor de entradas.....	48
5.6.3 Servidor.....	49
5.6.3.1 Gestor de comunicación.....	49
5.6.3.2 Generador de analíticas.....	53
5.6.3.3 Gestor de base de datos.....	55
5.6.4 Sistema de gestión.....	55
5.6.4.1 Controlador de la simulación.....	56
5.6.4.2 Graficador de resultados.....	58
5.6.5 Generador de reportes.....	59
5.6.5.1 Gestor de usuarios.....	60
5.6.6 Base de datos.....	61
5.6.7 Transmisión de la pantalla del Simulador.....	63
5.6.7.1 Transmisión de imagen.....	63
5.6.7.2 Transmisión de estados.....	65
5.6.7.3 Selección de la alternativa.....	66
<b>Capítulo 6: Conclusiones y trabajos futuros.....</b>	<b>68</b>
6.1 Limitaciones y trabajos futuros.....	70
<b>Bibliografía.....</b>	<b>71</b>
<b>Anexo I – Protocolo de entrevistas a los expertos.....</b>	<b>77</b>
Expertos en simulación.....	77
Expertos en capacitación.....	78
Expertos en física.....	79
<b>Anexo II – Definición de Escenarios.....</b>	<b>80</b>

## Capítulo 1: Introducción

La formación profesional se basa en que los aprendices adquieran los conocimientos, destrezas y aptitudes necesarios para el ejercicio de su profesión [1]. La incorporación de nuevas tecnologías en este ámbito ha propiciado el uso de recursos educativos más novedosos y prácticos que motivan la formación de los aprendices [2].

En particular, los videojuegos han demostrado ser de utilidad para que los aprendices adquieran las habilidades necesarias para desempeñarse en el ámbito profesional [3]. Este es uno de los objetivos que persiguen los Juegos Serios, videojuegos diseñados con fines no lúdicos, como lo son la enseñanza y el entrenamiento [4].

Algunos ejemplos de Juegos Serios son: Metalloman, el cual enseña a estudiantes de ingeniería conceptos y procesos fisiológicos [5]; Darfur is Dying, un videojuego de concienciación que simula un campo de refugiados en Darfur, Sudán [6]; Power Down the Zombies, cuyo objetivo es mejorar la toma de decisiones relativas al uso racional, eficiente y consciente de la energía [7].

Entre los distintos tipos de Juegos Serios se encuentran los Simuladores de Entrenamiento, los cuales buscan emular en cierto grado alguna situación del mundo real para ofrecer un entorno en el cual el aprendiz pueda entrenar. Estos sistemas son especialmente útiles cuando el entrenamiento en un terreno real resulta inviable, ya sea porque resulta muy costoso recrear dicha situación o por los riesgos inherentes a la actividad en cuestión [8].

Vox Maris es uno de los Simuladores de Entrenamiento disponibles en el mercado. Este sistema, desarrollado por la empresa marplatense Advenio Software, es utilizado para el entrenamiento de operadores del Sistema Mundial de Socorro y Seguridad Marítima (GMDSS) [9]. Otro ejemplo es el simulador de cirugías dentales desarrollado por Rhienmora et al. [10], el cual permite a estudiantes de odontología practicar cirugías en posturas correctas, combinando modelos 3D de dientes y herramientas con la vista del mundo real.

Si bien estas nuevos recursos resultan prometedores, es imprescindible definir métodos que permitan evaluar de forma efectiva el aprendizaje y las habilidades adquiridas por los aprendices a lo largo de las sesiones de simulación.

Las Analíticas de Aprendizaje surgen como una solución que permite recopilar y analizar los datos resultantes de las interacciones de los aprendices con el Simulador de Entrenamiento, lo que permite medir su desempeño y producir evidencias válidas del aprendizaje [11].

Actualmente, las Analíticas de Aprendizaje están siendo empleadas en diversas áreas para analizar y mejorar los procesos educativos. Esto puede verse ejemplificado en sistemas como tutores inteligentes, foros de discusión, páginas web, cursos on-line masivos abiertos (MOOCs), entre otros [12].

Los Simuladores de Entrenamiento están siendo cada vez más utilizados para la formación de profesionales. Sin embargo, muchas de las implementaciones existentes en la actualidad no brindan experiencias que resulten motivantes para los aprendices o carecen de métodos que permitan evaluar de forma objetiva el desempeño del aprendiz. En este contexto se ha desarrollado este trabajo.

## 1.1 Objetivos

El objetivo general del proyecto es llevar a cabo la puesta en valor de un Simulador de Entrenamiento para el uso de artillería antiaérea, incorporando mejoras en la simulación para aumentar la experiencia inmersiva, conjuntamente con la medición de datos y presentación de informes que permitan realizar una evaluación objetiva del desempeño del aprendiz.

En ese marco se establecieron los siguientes objetivos específicos:

- Analizar herramientas disponibles para el desarrollo de Simuladores de Entrenamiento.
- Identificar información relevante del proceso de aprendizaje de los aprendices sobre la base de sus datos de interacción en el Simulador de Entrenamiento.
- Construir una solución que proporcione información sobre el aprendizaje de los aprendices en el Simulador de Entrenamiento para poder realizar Analíticas de Aprendizaje.

## 1.2 Publicaciones vinculadas

- Lanzillotta, F. y Massa, S. M. (2018). Analíticas de aprendizaje en simuladores de entrenamiento. Jornadas de Investigación de la Universidad Nacional de Mar del Plata (Investigar UNMDP), Mar del Plata, Buenos Aires, Argentina.
- Lanzillotta, F. y Massa, S. M. (2019). Simuladores de Entrenamiento y Analíticas de Aprendizaje: una revisión sistemática. IV Jornadas Argentinas de Tecnología, Innovación y Creatividad (JATIC), Mar del Plata, Buenos Aires, Argentina.
- Massa, S. M., Lanzillotta, F. y Spinelli, A. T. (2019). La elicitación de Serious Games en un Simulador de Entrenamiento. VII Congreso Nacional de Ingeniería Informática - Sistemas de Información (CoNallSI), San Justo, Buenos Aires, Argentina.
- Lanzillotta, F., Massa, S. M. y Spinelli, A. T. (2020). Puesta en valor de un simulador de entrenamiento mediante la incorporación de experiencia inmersiva y analíticas de aprendizaje. XXII Workshop de Investigadores en Ciencias de la Computación (WICC), El Calafate, Santa Cruz, Argentina.
- Lanzillotta, F., Massa, S. M. y Spinelli, A. T. (2020). Learning analytics in training simulators: a systematic review of literature (no publicado). Congreso Bienal IEEE ARGENCON 2020, Resistencia, Chaco, Argentina.

## Capítulo 2: Marco teórico

### 2.1 Juegos Serios

Los Juegos Serios (Serious Games en inglés) son aquellos videojuegos cuyo propósito principal no es el entretenimiento, el disfrute o la diversión [13]. Los principios de diseño de videojuegos son utilizados con otros fines, como pueden ser la capacitación, el entrenamiento, la publicidad, la simulación, o la educación [4].

Zyda [14] define al Juego Serio como un desafío mental que se juega con una computadora, siguiendo reglas específicas, y hace uso del entretenimiento para promover la formación en diversos ámbitos. Los Juegos Serios hacen uso de la informática y la tecnología que se aplican en los videojuegos con la finalidad de colaborar en el aprendizaje de situaciones del mundo real [15].

El aporte de los videojuegos al aprendizaje se ve evidenciado en muchos aspectos. Los jugadores aprenden a obtener información de múltiples fuentes, a tomar decisiones rápidamente, a deducir las reglas del juego, a crear estrategias para superar obstáculos, a experimentar y comprender sistemas complejos, a trabajar colaborativamente [16]. Estas ventajas hacen de los Juegos Serios una gran oportunidad para enseñar distintas habilidades y situaciones del mundo real.

Los Juegos Serios le brindan al usuario una alta inmersión sensorial a través de constantes interacciones [14]. Gracias a esto, el jugador puede entrar en lo que se denomina estado flow o experiencia óptima, un estado en la que la concentración y la inmersión son elevadas [17].

Existe una enorme cantidad de Juegos Serios disponibles hoy en día. PeaceMaker simula el conflicto israelí-palestino, donde el jugador representa al líder de uno de los dos bandos y debe reaccionar a distintos eventos y tomar decisiones sociales, políticas y militares, con el objetivo de resolver el conflicto [18]. Raíces tiene como objetivo concienciar a los niños sobre las diferentes problemáticas de los pueblos originarios y ayudar a fortalecer la diversidad cultural argentina [19]. Pacific es un curso online en formato de videojuego, ambientado en una isla desierta en la que el jugador y su equipo han quedado atrapados

tras un accidente aéreo. Para escapar con vida, el jugador debe aprender y poner en práctica estrategias, técnicas y herramientas para desarrollar y mejorar sus habilidades de liderazgo y gestión de equipos [20].

## 2.2 Simuladores de Entrenamiento

Vidal-Gomel y Fauquet-Alekhine [21] definen a los simuladores como artefactos que simulan, de forma parcial o completa, el comportamiento o funcionamiento de un equipamiento, un sistema técnico, o un fenómeno natural. Algunos simuladores buscan igualar la apariencia y el comportamiento del sistema real para brindar una experiencia realista, mientras que otros se limitan a simular únicamente los aspectos clave de dicho sistema de una manera más abstracta [22].

Estos sistemas permiten sustituir la realidad por escenarios simulados en los que estudiantes y profesionales pueden entrenar para desarrollar diversas habilidades y adquirir nuevos conocimientos. Estos escenarios, y las metodologías que se aplican en ellos, varían según las habilidades a entrenar [23]. Esto brinda la posibilidad de que los aprendices se capaciten sin incurrir en los costos operativos de hacerlo en un terreno real y sin enfrentarse a los riesgos propios de esa actividad [8].

Coincidiendo con Zyda [14], los Simuladores de Entrenamiento pueden ser analizados como un caso particular de Juegos Serios, debido a que la interacción del usuario con el entorno simulado genera una alta inmersión sensorial. Por lo tanto, los estudios realizados en esta temática pueden ser aprovechados para el desarrollo de Simuladores de Entrenamiento.

Los Simuladores de Entrenamiento son especialmente utilizados en aquellos dominios donde se requiere que las personas manipulen o controlen sistemas complejos [22]. En la actualidad, están siendo ampliamente utilizados en diversos ámbitos, como el adiestramiento militar, el entrenamiento civil y las aplicaciones comerciales [24]. En el ámbito médico, por ejemplo, el uso de Simuladores de Entrenamiento ha reducido el riesgo asociado con errores humanos, lo que ha aumentado la seguridad de los pacientes [25].

En la actualidad existe una gran variedad de simuladores que son utilizados para entrenar y capacitar en diversos ámbitos. Microsoft Flight Simulator es una serie de

simuladores de vuelo que ayuda a aviadores civiles a realizar prácticas de vuelo [26]. Virtual Interactive Combat Environment es un simulador de entrenamiento táctico que le permite al personal militar adiestrarse en la resolución de conflictos, el uso de diversas armas, la comunicación con el equipo y la participación en escenarios de guerra [27]. El simulador de Puesto de Comando de Defensa Aérea FIERRO es un sistema que permite simular la planificación y ejecución en un teatro de operaciones de las diversas amenazas que pueden aparecer a la hora de defender un cierto espacio aéreo o territorio [28].

El entrenamiento basado en la simulación debe estar siempre acompañado de una etapa de retroalimentación, en la cual tanto los instructores como los aprendices analicen la actividad que se ha llevado a cabo, sus puntos fuertes y los aspectos a mejorar [23]. En este punto, es importante contar con medios que posibiliten dicho análisis. Es por esto que resultan necesarias herramientas que brinden información del desempeño del aprendiz en el entorno simulado.

## 2.3 Analíticas de Videojuegos

Las Analíticas de Videojuegos son el proceso de descubrir y comunicar patrones en los datos provenientes del videojuego y de sus jugadores, los cuales sirven de apoyo para la resolución de problemas, la toma de decisiones y la mejora del rendimiento [32].

Drachen, Seif El-Nasr y Canossa [32] proponen una clasificación para estas métricas en función del origen de los datos y el objetivo de su análisis:

- **Métricas del usuario:** son las que se relacionan con los usuarios desde dos perspectivas diferentes. La primera percibe a los usuarios como clientes y se utiliza al calcular métricas relacionadas con los ingresos. Por el contrario, la segunda perspectiva se enfoca en el comportamiento de los usuarios como jugadores en el videojuego.
- **Métricas de rendimiento:** son aquellas que se utilizan para monitorear el rendimiento de la infraestructura técnica y la infraestructura de software detrás de un videojuego. Estas métricas son especialmente usadas en el control de calidad.
- **Métricas del proceso:** son las que permiten realizar el seguimiento del proceso de desarrollo del videojuego.

De la misma forma, presentan una subclasificación específica para las métricas del usuario, argumentando que la gran mayoría del conocimiento sobre Analíticas de Videojuegos está basado en estas métricas [32]:

- **Métricas del cliente:** son aquellas relacionadas con los usuarios como clientes o fuentes de ingreso.
- **Métricas de la comunidad:** son las utilizadas para monitorear los movimientos de las personas en la comunidad de usuarios del videojuego.
- **Métricas del gameplay:** son las que se relacionan con las interacciones de los usuarios como jugadores dentro del videojuego.

Esta clasificación puede verse resumida en el diagrama de la Figura 1.



Figura 1. Diagrama jerárquico de las métricas del videojuego (adaptado de [32])

### 2.3.1 Analíticas de Aprendizaje

Las Analíticas de Aprendizaje han sido definidas en el marco de la International Conference on Learning Analytics and Knowledge (LAK '11) como la medida, colección,

---

análisis y presentación de datos sobre los aprendices y su contexto, con el propósito de comprender y optimizar el aprendizaje y los entornos en los que se produce. A esta definición, Ferguson [29] añade que que las Analíticas de Aprendizaje utilizan datos preexistentes que pueden ser leídos por una computadora y que sus técnicas pueden ser utilizadas para manejar grandes conjuntos de datos que no serían factibles de manejar manualmente.

Las Analíticas de Aprendizaje tienen su origen en los conceptos de inteligencia empresarial, el uso de datos en una empresa para facilitar la toma de decisiones. Estas ideas han sido apropiadas por las instituciones educativas con el objetivo de brindar nuevas herramientas que permitan comprender y optimizar el aprendizaje, así como los entornos en los que este tiene lugar [30].

Hershkovitz et. al. [31] señalan que las Analíticas de Aprendizaje han tenido un enorme progreso en los últimos años, debido a que se han ido incorporando métodos estadísticos avanzados y técnicas de aprendizaje automático. Esto ha permitido que se puedan capturar y analizar una mayor cantidad de datos, así como también ha mejorado la calidad de la información que se obtiene como resultado de dicho análisis.

### **2.3.2 Analíticas de Aprendizaje en Juegos Serios**

Las Analíticas de Aprendizaje en Juegos Serios tienen como objetivo obtener información relevante sobre su proceso de aprendizaje. A través de este análisis se pretende comprender la forma en la que el estudiante aprende y cómo se puede diseñar y adaptar el videojuego para mejorar sus resultados. Este proceso resulta de la combinación de los objetivos de aprendizaje de las Analíticas de Aprendizaje y de las tecnologías utilizadas en las Analíticas de Videojuegos [33].

Una sesión de juego, por más corta que sea, puede generar una gran cantidad de datos, ya que las secuencias de interacción y reacción dentro del videojuego suelen ser rápidas y muy frecuentes [34]. Sin embargo, la aplicación de las Analíticas de Aprendizaje en Juegos Serios no necesariamente requiere la obtención de una gran cantidad de datos, sino que los datos obtenidos sirvan para determinar el progreso del estudiante en el cumplimiento de los objetivos de aprendizaje [35].

Alonso Fernández [33] propone los siguientes pasos para convertir en analíticas los datos generados por los jugadores:

1. Recolectar las interacciones de los estudiantes en el Juego Serio como evidencia de su proceso cognitivo, habilidades y destrezas.
2. Analizar las secuencias de acciones obtenidas a través de técnicas de aprendizaje automático y/o estadístico.
3. Presentar visualizaciones de los resultados obtenidos.

Previo a comenzar a diseñar e implementar el videojuego, es necesario definir los objetivos de aprendizaje que se espera que los estudiantes logren a través de sus interacciones con el videojuego. Estos objetivos deben ser los lineamientos que guíen el desarrollo de todo el proceso de Analíticas de Aprendizaje en Juegos Serios, ya que determinarán tanto la forma en la que se analizarán los datos como los resultados que se mostrarán en las visualizaciones [33]. Este proceso se muestra en la Figura 2.

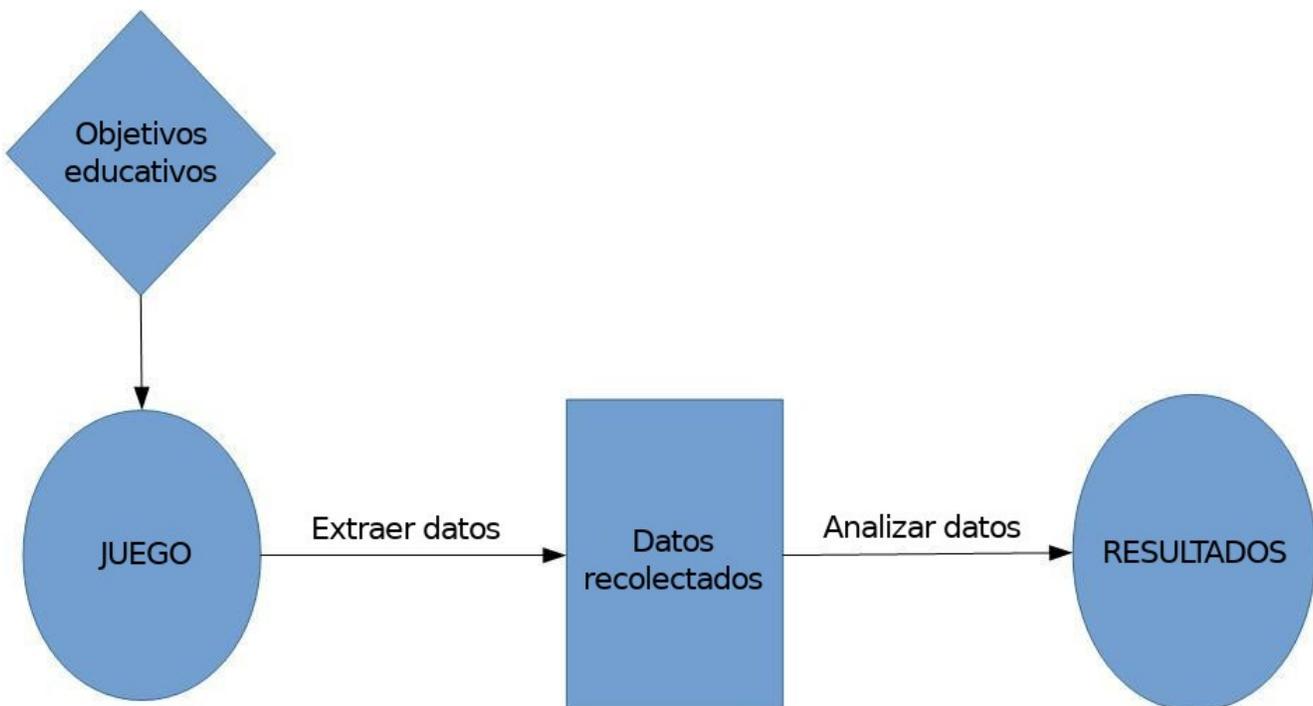


Figura 2. Proceso de las Analíticas de Aprendizaje en Juegos Serios (adaptado de [33])

## Capítulo 3: Estado del arte

### 3.1 Metodología

Con el objetivo de conocer qué han desarrollado otros autores respecto al uso de las Analíticas de Aprendizaje en los Simuladores de Entrenamiento, se ha trabajado con la metodología de Kitchenham [36] para el desarrollo de una revisión sistemática de la literatura. Esta técnica es una forma de identificar, evaluar e interpretar todas las investigaciones relevantes a una pregunta de investigación particular, una temática o un fenómeno de interés. Tiene como objetivo presentar la evolución de una temática de investigación a través de una metodología confiable, rigurosa y auditable.

La metodología de Kitchenham consta de las siguientes etapas:

1. Planificación de la revisión
  - a. Identificación de las necesidades que motivan la revisión
  - b. Definición de un protocolo de revisión
2. Procedimiento de la revisión
  - a. Identificación de la investigación
  - b. Selección de los estudios primarios
  - c. Evaluación de la calidad de los estudios
  - d. Extracción de datos y seguimiento
  - e. Síntesis de los datos y seguimiento
3. Discusión y conclusiones del estudio

## **3.2 Planificación de la revisión**

### **3.2.1 Identificación de las necesidades que motivan la revisión**

Actualmente, las Analíticas de Aprendizaje resultan de mucho interés a la hora de obtener resultados cuantitativos respecto del desempeño de los estudiantes o aprendices. Si bien su inclusión dentro de los Juegos Serios se encuentra en constante crecimiento, aún se ha trabajado poco sobre su uso puntual en los Simuladores de Entrenamiento. Respecto a estos Juegos Serios en particular, no se posee mucho conocimiento respecto a la aplicación de Analíticas de Aprendizaje en ellos.

Partiendo de estas ideas, se elaboraron las preguntas de investigación que han de ser respondidas a través de la revisión sistemática de la literatura, las cuales se encuentran detalladas a continuación:

- 1) ¿Existen iniciativas relacionadas con el uso Analíticas de Aprendizaje en Simuladores de Entrenamiento?
- 2) ¿Se han publicado metodologías de implementación de Analíticas de Aprendizaje en Simuladores de Entrenamiento?
- 3) ¿Existen investigaciones que utilicen las Analíticas de Aprendizaje en Simuladores de Entrenamiento para brindar recomendaciones y adaptar el sistema a las necesidades del aprendiz?
- 4) ¿Qué información resulta relevante respecto al desempeño del aprendiz en un Simulador de Entrenamiento?

### **3.2.2 Definición de un protocolo de revisión**

Con el objetivo de realizar las búsquedas de los artículos pertinentes a la revisión, se seleccionaron las bases de datos con incumbencias en las áreas de interés. Las bases de datos que han sido utilizadas son las siguientes: ACM, ERIC, IEEEExplore, ScienceDirect, Scopus y SpringerLink. A partir de las preguntas de investigación, se determinó la cadena de búsqueda a emplear:

*“simulator” AND “learning analytics”*

Se han establecido criterios para poder seleccionar, entre todos los resultados que se obtengan, los artículos apropiados para responder estas preguntas.

- **Criterios de inclusión:**

- Artículos publicados entre los años 2011 y 2018.
- Artículos escritos en inglés o español.
- Artículos accesibles.
- Artículos que relacionen los simuladores de entrenamiento con el uso de analíticas de aprendizaje.
- Artículos que presenten implementaciones de analíticas de aprendizaje en simuladores de entrenamiento.
- Artículos revisados por pares.

- **Criterios de exclusión:**

- Artículos publicados antes del año 2011.
- Artículos que no estén escritos en inglés o español.
  - Artículos no accesibles.
  - Artículos en los que únicamente se mencionen simuladores como ejemplo.
  - Artículos en los que el término “simulador” solo figure en las referencias.
  - Artículos en los que solo se utilice al simulador como herramienta para otro fin que no sea la capacitación.

### 3.3 Resultados de la revisión

Las búsquedas realizadas en las bases de datos seleccionadas han dado como resultado un total de 50 artículos. Luego de haber aplicado los criterios de inclusión y exclusión, se seleccionaron 6 artículos con los que se han respondido las preguntas de investigación.

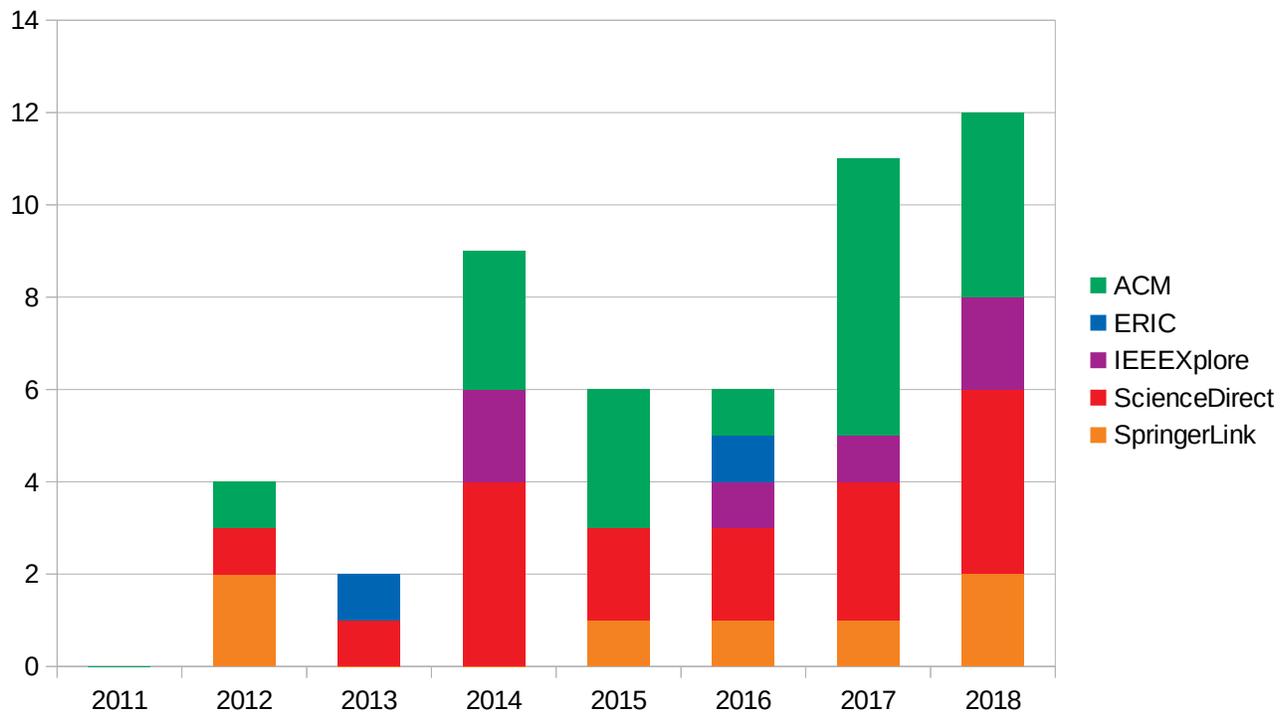


Figura 3. Cantidad de artículos encontrados en cada biblioteca en función del tiempo.

### 1) ¿Existen iniciativas relacionadas con el uso Analíticas de Aprendizaje en Simuladores de Entrenamiento?

El uso de Analíticas de Aprendizaje en el campo de los Simuladores de Entrenamiento es abordado de diversas formas en los estudios analizados.

El trabajo de Gonçalves et al. [37] se basa en realizar evaluaciones en las que se combinan prácticas de cálculo, laboratorios prácticos, simulaciones y laboratorios remotos. Sin embargo, las Analíticas de Aprendizaje son aplicadas únicamente sobre las respuestas de los estudiantes a las evaluaciones y no sobre la simulación en sí misma.

Tibola [38], por su parte, propone realizar Analíticas de Aprendizaje sobre un conjunto de herramientas destinadas al aprendizaje de estudiantes de ingeniería, entre las que se incluyen laboratorios prácticos desarrollados en mundos virtuales 3D. En este caso, la actividad en los laboratorios simulados, al igual que en el resto de las herramientas, es monitoreada a través de un panel de control que los docentes utilizan para evaluar.

Yin et al. [39] incorporan puntuación de resultados automatizada en un simulador de realidad virtual para cirugías endodónticas. Estos simuladores tienen la capacidad de registrar datos detallados de los movimientos que los usuarios realizan en cada paso de una tarea, lo cual brinda una gran cantidad de datos a partir de los cuales se puede evaluar el desempeño de los estudiantes.

Dodero et al. [40] exponen en su artículo la importancia de los requisitos no funcionales cuando se implementa un sistema de Analíticas de Aprendizaje. Esto se ve evidenciado en el caso práctico que abordan en dicha publicación, en el cual detallan el trabajo llevado a cabo para armonizar los requisitos de interoperabilidad y eficiencia para dar soporte a Analíticas de Aprendizaje en tiempo real en un simulador de entrenamiento naval.

La investigación de Xanthopoulos y Xinogalos [41] trata sobre juegos móviles basados en la ubicación en el ámbito educativo. Entre ellos incluyen a las simulaciones participativas, en las cuales los estudiantes exploran un entorno de realidad aumentada de forma colaborativa. A lo largo de este estudio, repasan tecnologías disponibles para incorporar Analíticas de Aprendizaje en estos sistemas. También señalan la necesidad de formalizar la estructura de los datos recopilados a fin de lograr un nivel de abstracción que reduzca la complejidad y facilite la interoperabilidad de estas tecnologías.

Por último, Andrade [42] presenta una propuesta en la cual hace uso de tecnologías de detección. Estas herramientas permiten capturar grandes cantidades de datos relacionados con el aprendizaje a través de múltiples modalidades, como los movimientos de las manos y la dirección de la mirada. Con el objetivo de explorar representaciones detalladas de los movimientos corporales de los estudiantes durante su interacción con la simulación, el autor utiliza Analíticas de Aprendizaje Multimodal. Esta técnica representa los comportamientos observables de los estudiantes a partir de los datos recopilados por las tecnologías de detección.

## **2) ¿Se han publicado metodologías de implementación de Analíticas de Aprendizaje en Simuladores de Entrenamiento?**

Dentro de los artículos tenidos en cuenta, no se han hallado metodologías que guíen la implementación de Analíticas de Aprendizaje de manera específica en Simuladores de Entrenamiento. En todos los casos prácticos, los autores se limitan a informar, en mayor o

menor medida, el trabajo que ellos han llevado a cabo, sin exponer un conjunto de pasos a seguir para esta tarea que pueda extrapolarse a otros proyectos.

Aún así, resulta relevante resaltar lo publicado por Dodero et al. [40], quienes explican de forma general cuáles los requisitos no funcionales que deben ser tenidos en cuenta a la hora de realizar la implementación de un sistema de Analíticas de Aprendizaje y por qué. Específicamente, hacen especial hincapié sobre dos requisitos que ellos consideran críticos: la interoperabilidad y el rendimiento. Respecto a estos, brindan algunas recomendaciones prácticas que pueden aportar a conseguir su cumplimiento.

### **3) ¿Existen investigaciones que utilicen las Analíticas de Aprendizaje en Simuladores de Entrenamiento para brindar recomendaciones y adaptar el sistema a las necesidades del aprendiz?**

Varios de los artículos tratados hablan sobre aprovechar los resultados arrojados por las Analíticas de Aprendizaje para que el sistema se retroalimente y pueda tomar acciones en consecuencia.

Xanthopoulos y Xinogalos [41] exponen la posibilidad de evaluar el progreso del jugador y adaptar el juego a las necesidades personales.

Por su lado, Gonçalves et al. [37] proponen la implementación de sistemas de recomendación que ofrezcan sugerencias destinadas a aumentar el rendimiento de los estudiantes en las actividades de aprendizaje.

En cambio, Yin et al. [39] tratan sobre la generación de retroalimentación formativa, la cual requiere la evaluación del resultado, la correlación del procedimiento con errores en el resultado y el uso de la información para comunicar la evaluación en un lenguaje natural. Estos autores mencionan que esta práctica de comunicación no está ampliamente disponible actualmente.

### **4) ¿Qué información resulta relevante respecto al desempeño del aprendiz en un Simulador de Entrenamiento?**

De los estudios seleccionados puede desprenderse que los resultados relevantes del desempeño del aprendiz en los Simuladores de Entrenamiento dependen en gran medida del dominio sobre el cual se esté trabajando. Sin embargo, de la información que brindan

pueden extraerse algunos lineamientos generales que pueden ser utilizados independientemente del dominio.

Yin et al. [39] indican que la comparación de los resultados de un novato con los de expertos en el área es una medida comúnmente utilizada para la evaluación. Además, resaltan la importancia de incluir información específica sobre los errores cometidos por el estudiante, como el tipo y la gravedad de los mismos.

Cuando se trabaja con datos de los movimientos de los usuarios, Andrade [42] señala que los patrones de dichos movimientos capturan la comprensión del estudiante de la dinámica del sistema. Adicionalmente, los datos de la mirada permiten monitorizar la atención de los aprendices a las distintas fuentes de información que se muestran en la pantalla.

A partir del artículo de Xanthopoulos y Xinogalos [41] pueden enumerarse algunas métricas que posiblemente sean de utilidad para quienes supervisan el desempeño en la simulación. Entre ellas se incluyen: el número de simulaciones ejecutadas, las duraciones de las mismas, la cantidad de veces que el ocurrió un determinado evento.

Por su parte, Doderó et al. [40] mencionan que al registrar cada evento que ocurra durante la simulación, se le da la posibilidad al instructor de poder reproducirlos para ver cómo reaccionan los estudiantes ante distintos eventos y evaluar sus tareas.

Finalmente, el estudio realizado por Gonçalves et al. [37] destaca que, más allá de que se le dé la posibilidad al alumno de mejorar resultados no satisfactorios, es importante preservar el historial de resultados con el fin de tener información sobre las dificultades que enfrentan los estudiantes.

### **3.4 Discusión y conclusiones del estudio**

En este estudio se ha llevado a cabo una SRL con el objetivo de determinar los fundamentos disponibles en la literatura sobre la integración de las LA en los TS.

A partir de los artículos hallados, se ha comprobado que, si bien la temática se encuentra en crecimiento, aún no existe mucha información disponible y son pocos los desarrollos que se han realizado. Además, todos estos artículos abordan la temática de forma práctica, sin

aportar a la definición un marco teórico respecto sobre la integración de estas dos herramientas.

Aún así, es importante destacar que, si bien es escaso lo que se ha encontrado, las implementaciones propuestas son muy distintas unas de otras y presentan una interesante variedad de dominios y soluciones para el uso de LA en TS. Algunos han llegado incluso a proponer que estos sistemas brinden recomendaciones y se adapten a las necesidades del aprendiz.

Por otro lado, si bien no se han expuesto metodologías para la implementación de estas soluciones, los desarrollos tratados en los artículos analizados pueden aportar a la elaboración de lineamientos generales que puedan ser utilizados en trabajos futuros. Asimismo, han brindado importantes referencias respecto a la información que resulta relevante sobre el desempeño del aprendiz.

## Capítulo 4: Metodología

El producto planteado para el trabajo experimental se basa en dos pilares fundamentales. En primer lugar, el desarrollo de un Simulador de Entrenamiento que brinde un entorno realista e inmersivo, en el cual el aprendiz pueda formar los conocimientos y habilidades necesarios para operar correctamente los instrumentos simulados. Por otro lado, la generación de informes sobre el proceso de aprendizaje a partir de los datos provenientes de las interacciones de los aprendices en el Simulador de Entrenamiento.

A partir de estas propuestas, se han definido las metodologías a seguir: el Modelo de Proceso de Desarrollo para Juegos Serios [43] para el desarrollo del Simulador de Entrenamiento, y el marco de trabajo GLEANER [44] para la vinculación de las Analíticas de Aprendizaje en dicho Juego Serio.

### 4.1 Modelo de Proceso de Desarrollo para Juegos Serios

El Modelo de Proceso de Desarrollo para Juegos Serios (MPDSG) [43] es una metodología para la construcción de Juegos Serios desarrollada por el Grupo de Investigación en Tecnologías Interactivas (GTI) de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata. El MPDSG resulta de la combinación del Modelo de Proceso para el desarrollo de Objetos de Aprendizaje (MPOBA) [45], el modelo de Líneas de Producción de Software (LPS) [46] y el Diseño Centrado en el Usuario (DCU) [47].

El MPDSG define tres fases a través de las cuales se ha de llevar a cabo el desarrollo de los Juegos Serios, siguiendo los lineamientos de Rogers [48]:

- 1) **Fase de Pre-producción:** se realiza el diseño conceptual y se comienzan a definir aspectos generales del Juego Serio.
- 2) **Fase de Producción:** se diseña y se elabora el Juego Serio, culminando finalmente con el lanzamiento al mercado del mismo.
- 3) **Fase de Post-producción:** se corrigen los errores, el Juego Serio es ajustado y mejorado.

El MPDSG propone un proceso de desarrollo iterativo en el cual se involucran a los distintos expertos que se encuentran relacionados con la producción del Juego Serio: los expertos en educación (docentes), los expertos del contenido o dominio y los expertos en jugabilidad (jugadores) [43]. En cada ciclo de este proceso se realiza una etapa de especificación, destinada a establecer los requerimientos y las características del producto. En esta etapa pueden identificarse tres fases [49]:

- **Elicitación:** se obtienen los requerimientos del videojuego, los cuales han de ser redactados de forma comprensible y no ambigua.
- **Especificación:** se elaboran, en base a los requerimientos, los documentos que describen el videojuego que se ha de desarrollar.
- **Validación:** los expertos involucrados validan los requerimientos obtenidos y los documentos redactados.

El MPDSG se detalla en la Figura 4.

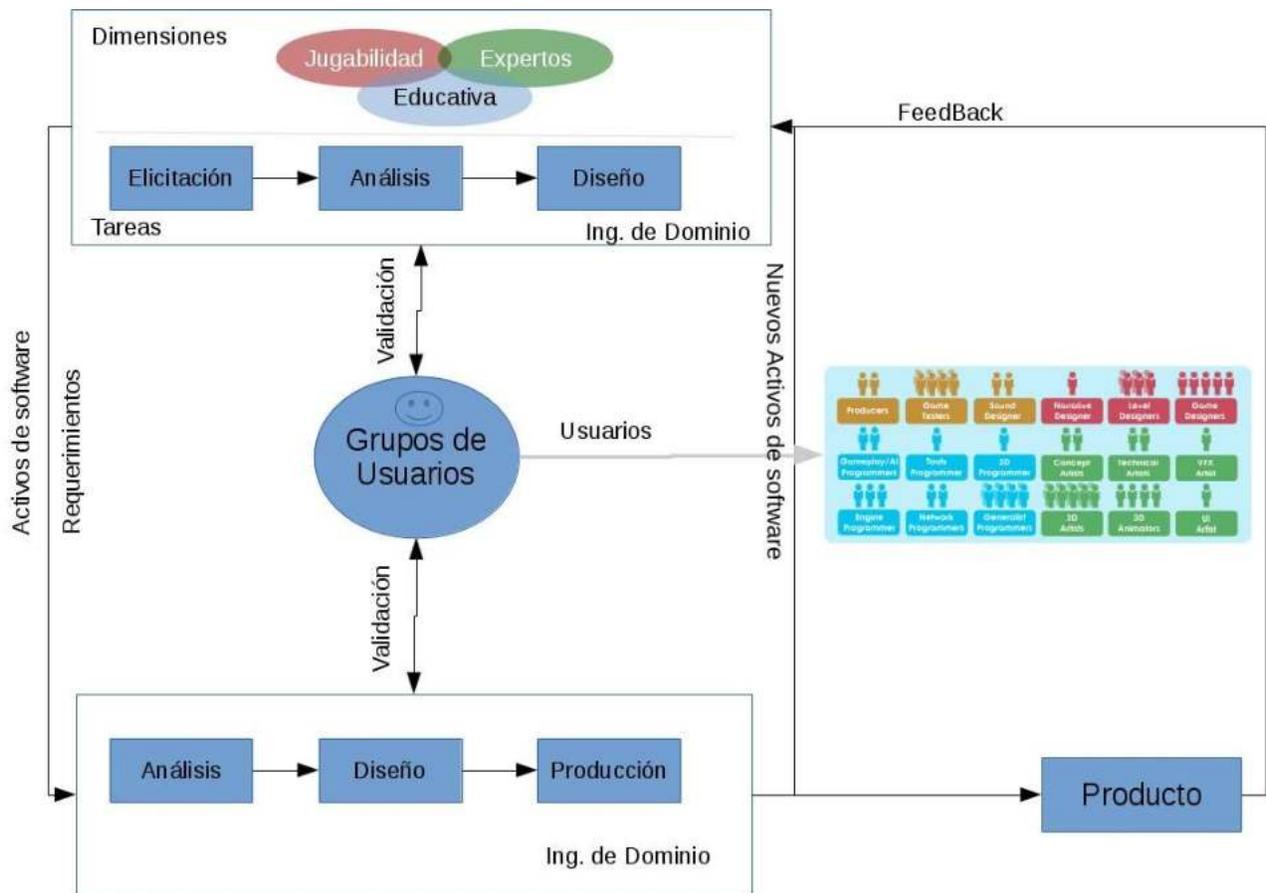


Figura 4. Modelo de Proceso de Desarrollo para Juegos Serios [43]

En la fase de elicitación es cuando se recopila información de los distintos actores relacionados con el desarrollo, con el objetivo de relevar los requerimientos de producto. Para ello se emplean diversas herramientas, tales como: entrevistas, cuestionarios, brainstorming, entre otras [49].

A partir de la información obtenida, se procede a realizar la construcción de los escenarios de Leite et al. [50]. Esta herramienta es utilizada para realizar la descripción de los requerimientos del videojuego de forma clara y concisa en lenguaje natural. Estos escenarios cuentan las acciones que pueden ocurrir dentro del videojuego, cuáles son los sujetos y objetos involucrados, qué cambios de estado pueden sufrir estos y cómo ocurren. Su estructura puede verse detallada en la Tabla 1.

*Tabla 1. Estructura del escenario de Leite [49]*

<b>Componente</b>	<b>Descripción</b>
<b>Título</b>	Titulo del escenario. Debe ser descriptivo del contenido (el requerimiento que escenifica).
<b>Objetivo</b>	Descripción del objetivo u objetivos que se busca cumplir con el requerimiento.
<b>Contexto</b>	Descripción de lugar, tiempo y precondiciones que se deben cumplir antes, durante el escenario y al final.
<b>Recursos</b>	Los objetos que participan en el escenario, sean físicos o virtuales.
<b>Actores</b>	Sujetos activos o pasivos que participan del escenario.
<b>Set de Episodios</b>	Lista de ejecución secuencial, de los episodios que ocurren en los escenarios. Los episodios pueden ocurrir en forma condicional. Se empieza por el primero y el escenario culmina con el último. Algunos episodios pueden ser escenarios en si mismos.
<b>Casos Alternativos</b>	Un escenario puede tener una situación no prevista que dispara acciones u otros escenarios. Por ejemplo, según alguna condicional la secuencia normal se altera (el escenario termina antes o tiene un comportamiento singular, por ejemplo).

De acuerdo con lo desarrollado por Spinelli et al. [51], cuando se trata de un Juego Serio en el cual se van a implementar Analíticas de Aprendizaje, resulta esencial definir el conjunto de habilidades a ser adquiridas por los aprendices a través de la extracción del conocimiento de los expertos. Luego, se deben relacionar con los escenarios construidos y, finalmente, determinar las variables a partir de las cuales se permite medir el alcance de estas habilidades. Todos estos elementos y relaciones debe ser validados por el juicio de los expertos y ajustarse a lo largo de las iteraciones del proceso de desarrollo.

## 4.2 Marco de trabajo para la vinculación de Analíticas de Aprendizaje en el diseño de Juegos Serios

Tal como señalan Westera, Nadolski y Hummel [52], el análisis de los datos del desempeño de los aprendices en los Juegos Serios posee dos diferentes etapas, las cuales se muestran en la Figura 5. En la primera, se analiza la información recopilada de cada usuario individualmente durante el juego, con el fin de mejorar y personalizar su interacción con el sistema. En la segunda etapa, posterior al juego, se trabaja con los datos de una población de jugadores para así poder evaluar y mejorar el videojuego en sí mismo y garantizar su calidad.

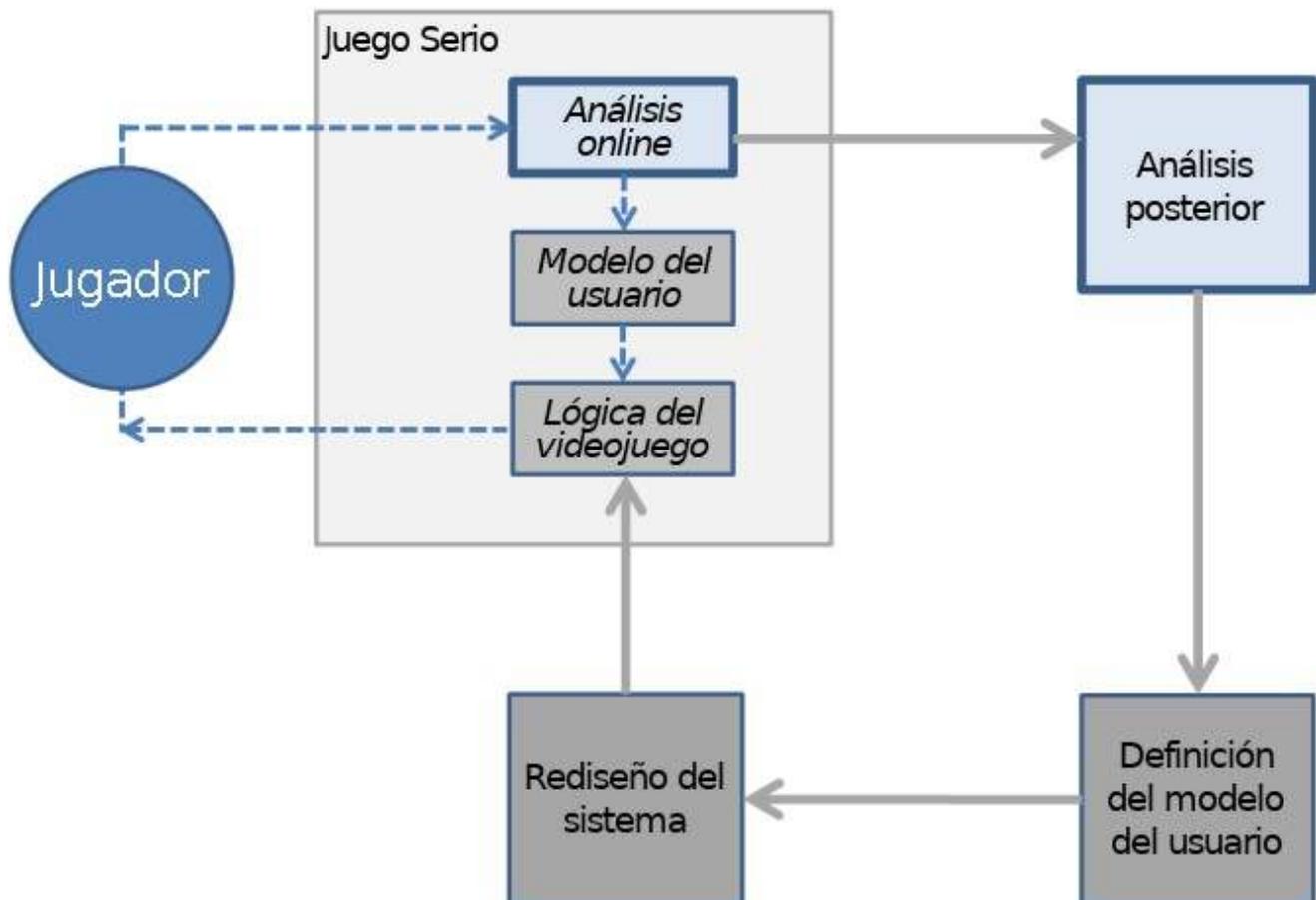


Figura 5. Análisis durante el juego y análisis posterior al juego (adaptado de [44])

De acuerdo con Baalsrud Hauge et al. [44], los tipos de datos a emplear en estas dos etapas son muy similares. Debido a esto, resulta imprescindible tener en cuenta la integración de las Analíticas de Aprendizaje desde la etapa de diseño de los Juego Serio. Su recomendación es definir e incluir en el sistema una capa que se encargue de traducir las acciones de los usuarios en información relevante para medir el comportamiento del jugador dentro del Juego Serio, para poder así analizar el cumplimiento de los objetivos de aprendizaje de acuerdo a los criterios de evaluación.

Estos autores han desarrollado, en conjunto con la Games and Learning Alliance (GALA), un marco de trabajo denominado GLEANER (Games and LEarning ANalytics for Educational Research) [44], el cual permite dar soporte al seguimiento y al análisis del comportamiento de los jugadores. La Figura 6 presenta las componentes principales de este marco de trabajo.



Figura 6. Componentes principales de GLEANER (adaptado de [44])

GLEANER está conformado por el Modelo de Analíticas de Aprendizaje (LAM), el cual define una secuencia de pasos y la información que se requiere en cada uno de ellos, y el Sistema de Analíticas de Aprendizaje (LAS), el cual implementa las funciones de procesamiento requeridas por el LAM.

GLENER es el resultado de una especificación del modelo descrito por Alonso Fernández [33] para el desarrollo de Analíticas de Aprendizaje en Juegos Serios. Los pasos definidos en dicho modelo son ampliados y profundizados, a partir de lo cual se define una serie de componentes que se encargan de realizar cada una de las tareas. De esta especificación surgen los siguientes pasos [53]:

1. **Selección y captura de los datos:** elegir y obtener los datos relevantes
2. **Agregación de los datos:** clasificar los datos obtenidos
3. **Presentación de informes de los datos:** convertir los datos en información
4. **Evaluación de los datos:** comprender la información reportada y convertirla en conocimiento utilizado para evaluar a los estudiantes
5. **Adaptación del juego:** adaptar el sistema en función de la evaluación.

## **Capítulo 5: Trabajo experimental**

### **5.1 Problema a resolver**

Ciertas actividades profesionales requieren de un arduo entrenamiento previo para poder operar en un terreno real. Estos adiestramientos pueden ser muy costosos o implicar grandes riesgos, tanto para quien lo realiza como para otras de las personas involucradas. Por lo tanto, son necesarias soluciones que minimicen estas variables y permitan capacitaciones acordes a la formación que se desea obtener.

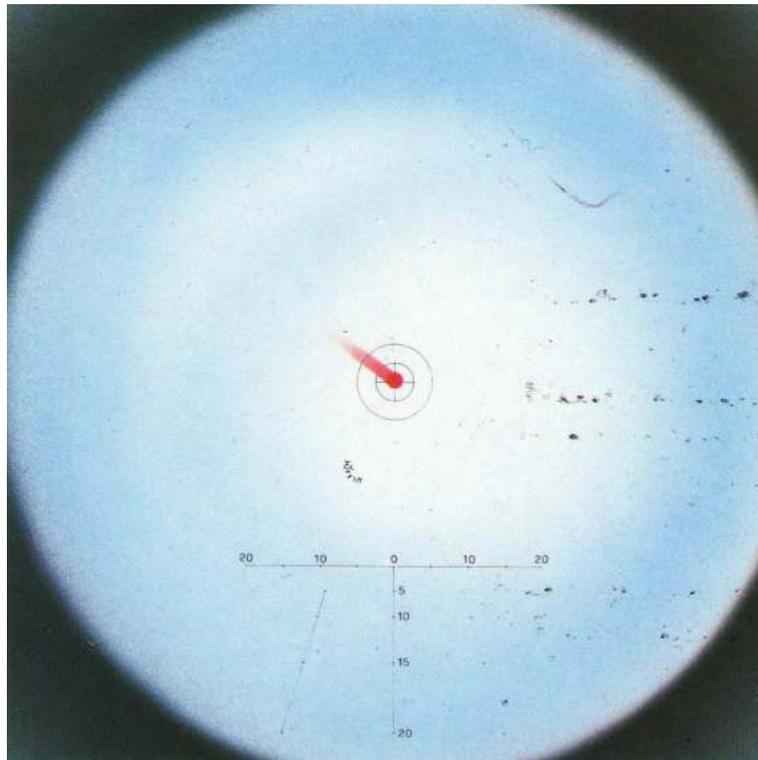
La Base Aérea Militar de la ciudad de Mar del Plata cuenta con un simulador que se utiliza para el entrenamiento en el uso de artillería antiaérea. El simulador original consistía en una pantalla de tubo rayos catódicos, montada sobre un cañón real, la cual era visualizada a través de la mira de dicho cañón. El cañón simulado era manejado con el control del cañón real.

En la pantalla se proyectaba un punto, el cual simulaba ser un avión. Este seguía una trayectoria recta, paralela al suelo, la cual era configurada a través de un control físico que permitía establecer sus parámetros. Este control puede verse en la Figura 7.



*Figura 7. Control de configuración del simulador original*

Los aviones aparecen en la simulación de uno en uno, pasando al siguiente cuando el avión actual era derribado. Este sistema tenía la posibilidad de que los aviones se dibujen sobre la imagen real que se ve a través de la mira. Sin embargo, esta función era muy poco utilizada. La Figura 8 muestra la visión del operador a través de la mira del cañón.



*Figura 8. Visión del operador en el simulador original*

Hace unos años se ha realizado una actualización del simulador, cambiando la pantalla original por una computadora y una pantalla a color. La electrónica original del control del cañón se adaptó para que dicha computadora pueda recibir su estado y la simulación pueda ser controlada por él. El control con el que se configura la simulación pasó de ser un elemento físico a controlarse a través de una computadora con un software desarrollado para dicho fin. El avión dejó de ser un punto para ser un triángulo de color rojo y el fondo pasó a ser una foto panorámica. La Figura 9 muestra la interfaz de este simulador actualizado.

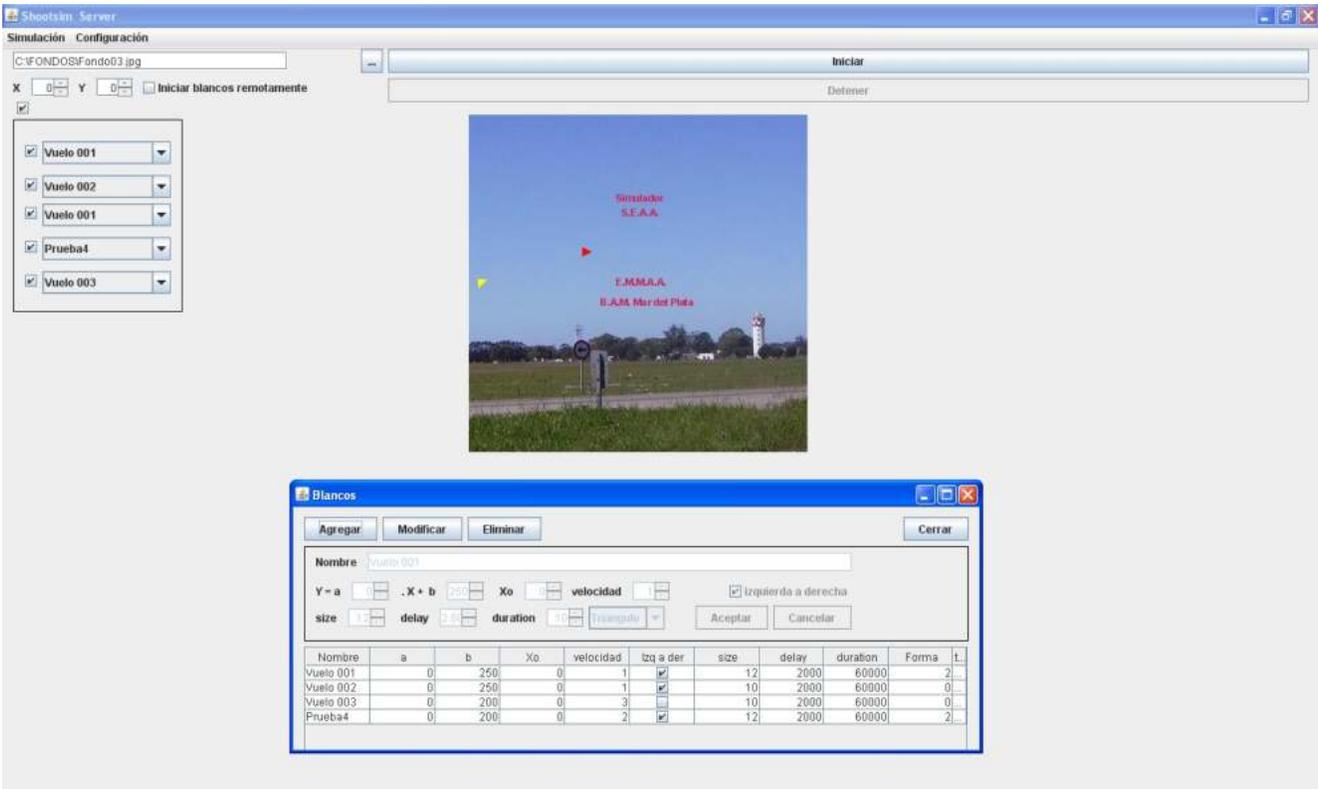


Figura 9. Primera actualización del simulador

Si bien esta herramienta les permite realizar entrenamientos en un entorno controlado y seguro, actualmente presenta grandes carencias. En primera instancia, la simulación ofrecida por el sistema es poco realista, lo que minimiza la inmersión del aprendiz en el entorno y desalienta su uso como herramienta de adiestramiento. Sumado a esto, la evaluación del aprendizaje y de las habilidades adquiridas carece de métodos objetivos y sistemáticos, ya que el instructor solo cuenta con la información que obtiene al observar la simulación y debe basarse exclusivamente en su criterio.

## 5.2 Dominio del problema

El cañón antiaéreo objetivo de la simulación es un cañón automático de doble tubo. Es capaz de rotar libremente de forma horizontal y también puede hacerlo hacia arriba y hacia abajo. Posee una mira a través del cual se puede observar a lo que el cañón está apuntando. A la hora de disparar, se presiona el pedal de disparo y la bala sale en la

dirección apuntada por la mira. Sin embargo, debido al tiempo que demora la bala en llegar a su objetivo y el desplazamiento que pueda realizar el avión en ese tiempo, el impacto difiere del esperado. Es por esto que el cañón posee una función de predicción que, al proveerle datos sobre la trayectoria del avión, corrige automáticamente esta desviación. Este cañón puede verse en la Figura 10.



*Figura 10. Cañón Oerlikon GAI-D01*

Al operador que maneja el cañón se le asigna un sector de responsabilidad, medido en grados. En todo momento debe mantener la mira del cañón dentro de este sector y nunca debe efectuar disparos fuera del mismo.

Cuando un avión es divisado, el operador debe centrarlo en la mira del cañón para identificarlo y realizar su seguimiento. La identificación de una aeronave se hace a través de su silueta, la cual permite distinguir si se trata de un avión amigo o enemigo. Luego, si se le da la orden de disparo, el operador debe derribar al objetivo en cuestión. Mientras el avión se mueve por zonas enemigas, realiza constantemente maniobras evasivas, con el fin de evitar ser derribado. Por lo tanto, es importante que el operador intente mantenerlo siempre en la mira.

## 5.3 Elicitación

La primera fase del MPDSG consiste en recopilar la información relevante para el desarrollo del sistema. En este caso, se optó por realizar entrevistas a los distintos expertos en el dominio del problema. Esta elección se debió a que con este método se formulan preguntas que, orientadas correctamente, permiten obtener de forma completa la información requerida. Por otro lado, da la posibilidad a los entrevistados de explayarse en sus respuestas, lo cual puede brindar, a veces, más detalle sobre el dominio, o brindar respuestas a preguntas que no se habían formulado hasta el momento.

Con el objetivo de focalizar las entrevistas, se identificaron y clasificaron a los expertos de acuerdo a su conocimiento en distintos aspectos del dominio:

- **Expertos en simulación:** aquellos con conocimiento sobre los aspectos técnicos del simulador y el hardware a emplear.
- **Expertos en capacitación:** quienes se encargan de instruir y evaluar a los aprendices en el uso de las herramientas que se pretenden simular.
- **Expertos en física:** brindan información acerca de la física y los datos técnicos de los elementos simulados.

Para cada grupo se elaboraron entrevistas dirigidas a reunir la mayor cantidad de información relevante posible, de acuerdo a su campo de conocimiento. Los protocolos que se han seguido a la hora de realizar estas entrevistas pueden verse en el Anexo I.

## 5.4 Especificación

La información obtenida en la elicitación ha sido utilizada en la siguiente fase para obtener los requerimientos y definir las características del producto.

### 5.4.1 Definición de los requerimientos

Los escenarios de Leite han sido confeccionados, validados por los expertos y mejorados a lo largo de las iteraciones. Las versiones finales de estos escenarios se muestran en Anexo II. Los requerimientos que surgen de esta especificación son los siguientes:

- El instructor configurará la simulación para adaptarla a las necesidades del entrenamiento a través de la especificación de ciertos parámetros, tanto de la simulación como de los aviones que se añadan a la misma.
- El simulador ofrecerá dos modos de entrenamiento. En el modo de seguimiento, el operador deberá apuntar al avión objetivo e intentar mantenerlo en la mira todo el tiempo que le sea posible, sin efectuar ningún disparo. Por su parte, el modo de tiro tendrá como objetivo que el operador logre apuntar y derribar a su avión objetivo en el menor tiempo posible.
- El operador interactuará con el simulador a través de los controles del cañón: podrá desplazar la mira, disparar y activar o desactivar la función de predicción.
- Los aviones aparecerán y se moverán en la simulación de acuerdo a los parámetros que se les han configurado, pudiendo ser derribados en cualquier momento por los disparos del cañón.
- El operador intentará siempre mantener la mira del cañón dentro del sector de responsabilidad que se le ha asignado en la simulación.
- El instructor observará en tiempo real la visión del operador junto con información y visualizaciones complementarias que le permitan saber lo que ocurre en todo momento dentro de la simulación.
- El sistema mostrará los resultados obtenidos durante la simulación, lo que le permitirá al instructor evaluar el desempeño del operador.
- El sistema generará reportes con los resultados de las simulaciones realizadas por el operador a través del tiempo, lo que le permitirá al instructor evaluar el progreso del operador a lo largo de las sesiones de simulación.
- El sistema ofrecerá un gestor de usuarios en el cual el administrador podrá agregar, editar, activar, desactivar y eliminar usuarios. Los usuarios podrán ser instructores, operadores y administradores.

## 5.4.2 Selección de Analíticas de Aprendizaje

Por otro lado, se definieron las habilidades a ser adquiridas por quienes utilicen el simulador, las cuales se detallan en las Tablas 2 y 3. Para cada habilidad se han determinado los escenarios asociados y se han seleccionado las Analíticas de Aprendizaje a través de las cuales se evalúa el desarrollo de la habilidad.

*Tabla 2. Habilidad - Realizar el seguimiento de objetivos*

Realizar el seguimiento de objetivos	
Escenarios asociados	Analíticas de Aprendizaje
<ul style="list-style-type: none"> <li>• <b>Buscar objetivo</b></li> <li>• <b>Seguir objetivo</b></li> </ul>	<ul style="list-style-type: none"> <li>• Porcentaje de seguimiento</li> </ul>
<ul style="list-style-type: none"> <li>• <b>Respetar sector de responsabilidad</b></li> </ul>	<ul style="list-style-type: none"> <li>• Porcentaje de responsabilidad</li> </ul>

*Tabla 3. Habilidad - Derribar a los objetivos enemigos*

Derribar a los objetivos enemigos	
Escenarios asociados	Analíticas de Aprendizaje
<ul style="list-style-type: none"> <li>• <b>Buscar objetivo</b></li> </ul>	<ul style="list-style-type: none"> <li>• Porcentaje de seguimiento</li> </ul>
<ul style="list-style-type: none"> <li>• <b>Derribar objetivo</b></li> </ul>	<ul style="list-style-type: none"> <li>• Tiempo en hacer blanco</li> <li>• Cantidad de disparos realizados</li> </ul>
<ul style="list-style-type: none"> <li>• <b>Respetar sector de responsabilidad</b></li> </ul>	<ul style="list-style-type: none"> <li>• Porcentaje de responsabilidad</li> <li>• Porcentaje de disparos responsables</li> </ul>

A continuación se describen las Analíticas de Aprendizaje seleccionadas:

- **Porcentaje de seguimiento:** mide en cada instante de tiempo, de 0 a 100, qué tan centrado está el avión en la mira del cañón. El valor 100 representa que el avión se encuentra en el centro de la mira, mientras que el valor 0 indica que el avión se

encuentra fuera del campo de visión del cañón. Se eligieron como valores representativos el promedio y el máximo a lo largo del tiempo.

- **Porcentaje de responsabilidad:** mide en cada instante de tiempo, de 0 a 100, qué tan alejado se encuentra el cañón de su sector de responsabilidad. El valor 100 representa que el cañón se encuentra dentro del sector, mientras que el valor 0 indica que el cañón se encuentra fuera del sector en el ángulo más alejado del mismo. Se eligieron como valores representativos el promedio y el mínimo a lo largo del tiempo.
- **Tiempo en hacer blanco:** indica el tiempo que el operador ha demorado en derribar el avión desde que este ha ingresado en su sector de responsabilidad.
- **Cantidad de disparos realizados:** cuenta la cantidad de disparos que el operador ha hecho mientras el avión se encontraba en la simulación.
- **Porcentaje de disparos responsables:** representa el porcentaje de los disparos totales que se han realizado dentro del sector de responsabilidad.

### 5.4.3 Parametrización de la simulación

Antes de dar comienzo a la simulación, es necesario que el instructor la configure para adaptarla a las necesidades del entrenamiento. Es por eso que se han analizado y determinado los parámetros a definir antes de cada ejecución. En primer lugar, se definieron las variables generales a toda la simulación:

- **Sector de responsabilidad:** define, en grados, el sector de responsabilidad asignado al cañón en el contexto de la simulación.
- **Modo de simulación:** si bien este parámetro no tiene efectos sobre la simulación en sí misma, define cuál es el objetivo de ese entrenamiento, lo cual permite clasificar los resultados a la hora de ser visualizados por los usuarios. Actualmente, los dos modos disponibles son **Seguimiento** y **Tiro**.

A su vez, ha sido necesario elegir qué conjunto de valores definen las trayectorias de los aviones añadidos a la simulación. Luego de analizarlo en conjunto con los expertos, se ha decidido, en una primera etapa, que los aviones describan trayectorias rectas paralelas al suelo, las cuales quedaron determinadas como se observa en la Figura 11.

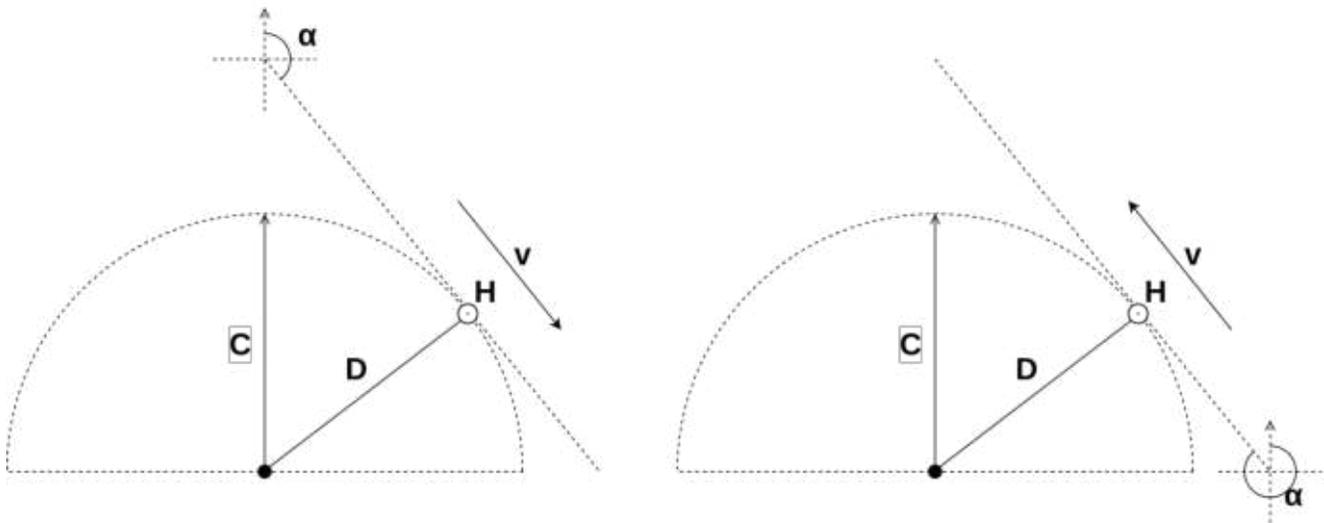


Figura 11. Parametrización de las trayectorias de los aviones

- **Distancia (D):** es la mínima distancia al cañón, medida en el plano horizontal, a la que el avión pasará.
- **Ángulo ( $\alpha$ ):** es el ángulo que se encuentra rotada la trayectoria respecto de la dirección inicial de la mira (C).
- **Altura (H):** es la altura a la cual vuela el avión, la cual permanece fija durante toda la trayectoria del mismo.
- **Velocidad (v):** es la velocidad a la cual se mueve el avión, la cual permanece fija durante toda la trayectoria del mismo.

Además, se han incluido dos parámetros opcionales adicionales que hacen crecer enormemente la variedad de situaciones que se pueden representar:

- **Tiempo:** es el momento de la simulación en el cual el avión aparece y comienza su trayectoria. Por defecto, esta variable se incrementa 10 segundos por cada nuevo avión que se agrega a la simulación. Sin embargo, el instructor puede definir el valor que desee, el cual puede incluso repetirse en distintos aviones.
- **Evaluar:** al existir la posibilidad de que haya varios aviones simultáneamente en la simulación, es imposible que el operador realice acciones sobre todos ellos. Es por esto que este parámetro define si las acciones del operador sobre el avión en

cuestión serán evaluadas o no. Es decir, si las analíticas que se generen serán o no incluidas en los reportes.

## 5.5 Diseño

En la fase de diseño, la especificación se ha analizado y, como resultado, se han definido la arquitectura, las componentes y el flujo de datos del sistema.

### 5.5.1 Arquitectura del sistema

La arquitectura seleccionada para el desarrollo del sistema ha quedado constituida por los componentes que se muestran en la Figura 12 y la distribución física descrita en la Figura 13.

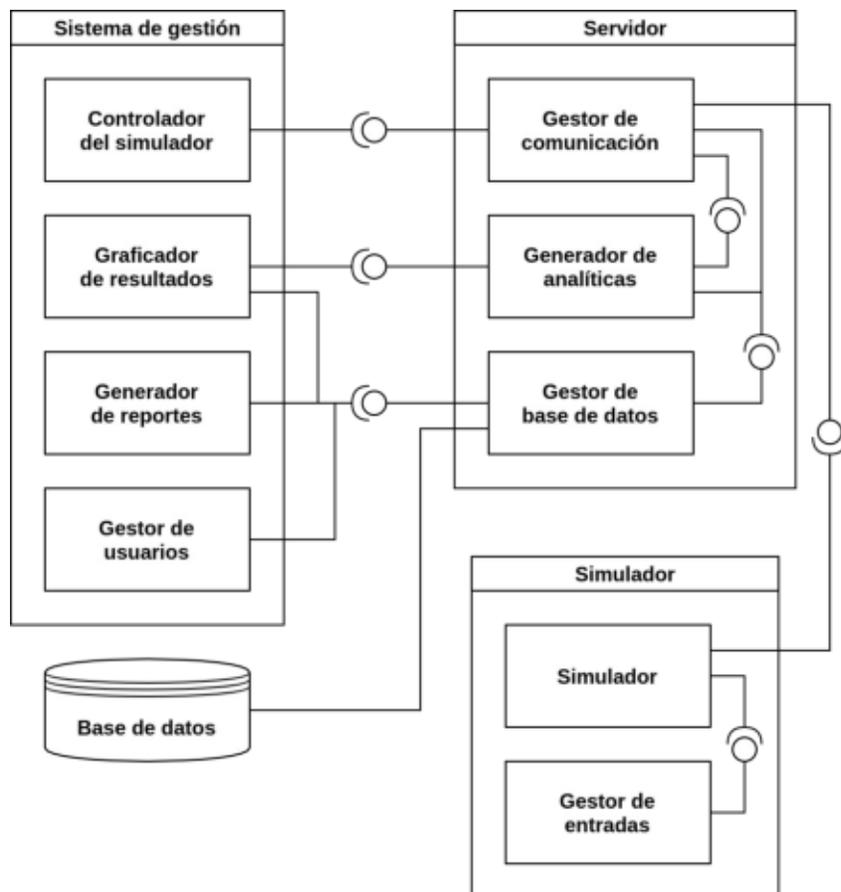


Figura 12. Diagrama de componentes del sistema

- **Simulador:** es el sistema en el cual se realiza el entrenamiento.
  - **Simulador:** es la componente que se encarga de ejecutar las simulaciones. Recibe las configuraciones necesarias para cada simulación, junto con los parámetros de los aviones. Permanentemente envía al *Servidor* todos los cambios de estado que ocurren dentro de la simulación.
  - **Gestor de entradas:** es responsable de leer las entradas físicas del control del cañón y enviarlas al *Simulador* para que sean representadas en la simulación.
- **Servidor:** es responsable de establecer el medio de comunicación entre las componentes y realizar la gestión de todos los datos del sistema.
  - **Gestor de comunicación:** es el medio a través del cual el *Controlador de la simulación* y el *Simulador* pueden comunicarse.
  - **Generador de analíticas:** se encarga de calcular e informar todos los valores de las analíticas asociadas a una simulación.
  - **Gestor de base de datos:** es la componente que se encarga de leer y escribir la información almacenada en la base de datos.
- **Sistema de gestión:** es a través del cual los usuarios pueden interactuar con todo el sistema.
  - **Controlador de la simulación:** permite al instructor configurar la simulación, agregar aviones y definir sus parámetros. Además, durante la simulación, muestra en tiempo real todo lo que está ocurriendo en la misma.
  - **Graficador de resultados:** muestra gráficos detallados de las analíticas a lo largo del tiempo de la simulación.
  - **Gestor de reportes:** muestra, en forma de gráficos, el progreso de los valores de las analíticas de los operadores a lo largo de las simulaciones.
  - **Gestor de usuarios:** permite administrar a los usuarios del sistema.

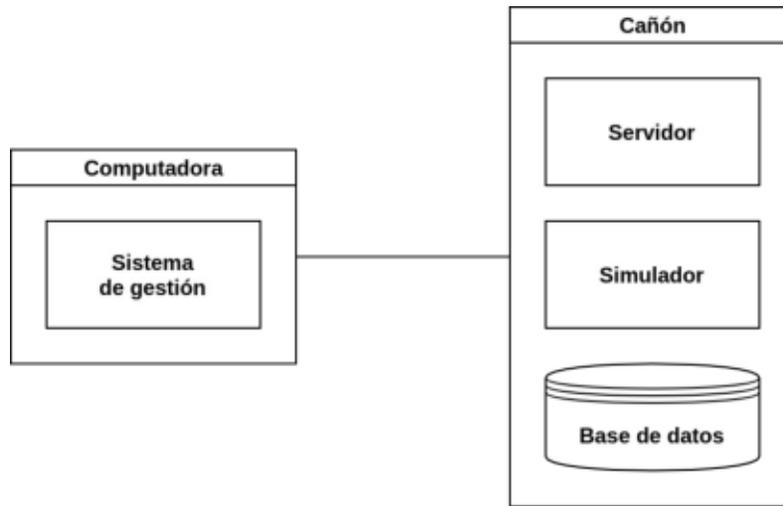


Figura 13. Diagrama de despliegue del sistema

Tanto el *Servidor* como el *Simulador*, junto con la *Base de datos*, están físicamente alojados en la computadora que se encuentra montada sobre un cañón real desarmado. El operador utiliza los controles propios de este cañón para manejar la simulación. Se ha convenido con el demandante el uso de una Raspberry Pi 3 modelo B para llevar a cabo esta tarea.

En tanto, se ha decidido que el *Sistema de gestión* sea un sitio web al cual los usuarios puedan acceder a través de cualquier computadora que se encuentre conectada por red a la computadora del cañón.

### 5.5.2 Analíticas de Aprendizaje

La vinculación de las Analíticas de Aprendizaje en este sistema se ha realizado siguiendo los lineamientos de GLEANER. Los pasos de este marco de trabajo han sido aplicados en la arquitectura del sistema de la siguiente forma:

1. El *Simulador* captura los eventos y las interacciones del usuario y envía estos datos al *Servidor*.
2. El *Generador de analíticas* del *Servidor* se encarga de calcular las Analíticas de Aprendizaje a partir de los datos enviados por el *Simulador*.

3. El *Graficador de resultados* muestra gráficos detallados de las Analíticas de Aprendizaje a lo largo del tiempo de la simulación. En tanto, el *Gestor de reportes* muestra el progreso de los valores de las analíticas de los operadores a lo largo de las simulaciones.
4. Debido a los alcances de este trabajo, el sistema no realiza evaluación alguna, si no que es utilizado por los instructores para que ellos sean quienes evalúen el desempeño de los aprendices.
5. De la misma manera, el *Simulador* no se adapta de forma automática en función de la evaluación, si no que son los instructores quienes configuran las simulaciones de acuerdo al progreso de los aprendices.

## 5.6 Producción

Durante la fase de producción se han seleccionado las tecnologías a emplear y se ha implementado el sistema de acuerdo a las definiciones de la fase de diseño.

### 5.6.1 Selección de motores de videojuegos

Siguiendo los lineamientos de Evans, Mileta y Petrillo [54], se ha llevado a cabo un análisis de las herramientas disponibles para el desarrollo de videojuegos. Se han evaluado distintos motores de videojuegos de propósito general disponibles en el mercado, con el objetivo de elegir el más apto para el desarrollo de este simulador de entrenamiento en particular. Los criterios utilizados para la evaluación han sido:

- Especificaciones básicas.
- Características excluyentes.
- Características deseables.
- Funcionalidades extras.
- Manejo de recursos de terceros.
- Licencias.
- Capacitaciones.

Particularmente, se seleccionaron las características a analizar:

- Características excluyentes:
  - Los videojuegos deben funcionar en la Raspberry Pi utilizada para este proyecto.
  - Manejo de modelos 3D.
  - Soporte para agregar distintas texturas a los objetos.
  - Empleo de física básica y colisiones.
  - Manejo de distintos protocolos de red (TCP, UDP, WebSocket, HTTP).
- Características deseables:
  - Facilidad para migrar el videojuego a otras plataformas.
  - Manejo de sistemas de partículas.
  - Soporte para agregar y configurar distintos materiales en los objetos.
  - Manejo de distintas condiciones de iluminación y sombras.

Las fuentes de datos consultadas han sido:

- <https://unity.com/>
- <https://docs.unity3d.com/>
- <https://forum.unity.com/>
- <https://www.unrealengine.com/>
- <https://docs.unrealengine.com/>
- <https://godotengine.org/>
- <https://docs.godotengine.org/>

Para realizar la evaluación, los criterios utilizados han sido ponderados de acuerdo a su importancia en el proyecto. También, se determinaron las características relevantes de cada uno de ellos, las cuales también fueron ponderadas en función de su peso relativo dentro de sus respectivos criterios. Estas características han sido puntuadas en forma de porcentaje

para cada uno de los productos evaluados. Luego, se han calculado los totales parciales de cada uno de los criterios y, finalmente, el total general para cada producto. A partir de los resultados obtenidos, se ha determinado que el motor de videojuegos Godot es el más apto para las necesidades del proyecto. El detalle de esta evaluación puede verse en la Tabla 4.

*Tabla 4. Evaluación de los motores de videojuegos*

	Características	Coficiente	Unity3D	Unreal	Godot
<b>Especificaciones básicas</b>	Lenguaje de codificación	0	C#	C++	GScript, C#
	Entorno de desarrollo (IDE)	50	80 %	70 %	100 %
	Plataformas soportadas	30	100 %	100 %	100 %
	Trabajo grupal	20	50 %	75 %	100 %
	<b>TOTAL</b>	<b>10</b>	<b>80 %</b>	<b>80 %</b>	<b>100 %</b>
<b>Características excluyentes</b>	Funcionamiento en la Raspberry Pi	30	100 %	100 %	100 %
	Manejo de modelos 3D	20	100 %	100 %	100 %
	Soporte para texturas	10	100 %	100 %	100 %
	Física básica y colisiones	20	100 %	100 %	100 %
	Manejo de protocolos de red	20	75 %	50 %	100 %
	<b>TOTAL</b>	<b>30</b>	<b>95 %</b>	<b>90 %</b>	<b>100 %</b>
<b>Características deseables</b>	Migración a otras plataformas	40	100 %	100 %	100 %
	Manejo de sistemas partículas	30	100 %	100 %	100 %
	Soporte para materiales	10	100 %	100 %	100 %
	Manejo de iluminación y sombras	20	100 %	100 %	100 %
	<b>TOTAL</b>	<b>10</b>	<b>100 %</b>	<b>100 %</b>	<b>100 %</b>
<b>Funcionalidades extras</b>	Codificación gráfica	10	75 %	100 %	100 %
	Profile	20	25 %	100 %	100 %
	Documentación y ejemplos	30	100 %	50 %	100 %
	Foros	30	100 %	100 %	100 %
	Soporte	10	100 %	100 %	75 %
	<b>TOTAL</b>	<b>5</b>	<b>78 %</b>	<b>85 %</b>	<b>98 %</b>
<b>Recursos de terceros</b>	Tienda de recursos	30	100 %	100 %	100 %
	Instalación automática desde GitHub	30	0 %	0 %	100 %
	Empaquetamiento de recursos e instalación manual	40	100 %	0 %	100 %
	<b>TOTAL</b>	<b>5</b>	<b>70 %</b>	<b>30 %</b>	<b>100 %</b>
<b>Licencias</b>	Costos	50	Gratis hasta US\$ 100 mil	Gratis hasta US\$ 3 mil	Gratis
	Posible variación del licenciamiento	50	Sí	Sí	No
	<b>TOTAL</b>	<b>20</b>	<b>50 %</b>	<b>50 %</b>	<b>100 %</b>
<b>Capacitaciones</b>	Conocimiento actual del estudiante	50	50 %	0 %	75 %
	Uso en la industria local	20	75 %	0 %	25 %
	Posibilidades de capacitación	30	75 %	25 %	50 %
	<b>TOTAL</b>	<b>20</b>	<b>63 %</b>	<b>8 %</b>	<b>58 %</b>
<b>TOTAL GENERAL</b>			<b>76 %</b>	<b>62 %</b>	<b>91 %</b>

## 5.6.2 Simulador

### 5.6.2.1 Simulador

El *Simulador* ha sido íntegramente desarrollado en Godot, un motor de videojuegos 2D y 3D multiplataforma, de código abierto publicado bajo la Licencia MIT y desarrollado por la comunidad de Godot [55]. Específicamente, se ha utilizado la versión 3.2 de dicho motor. Debido a que Godot no posee soporte para exportar los proyectos para la arquitectura del procesador de la Raspberry Pi utilizada, se ha optado por exportar al *Simulador* a HTML5 y ejecutarlo en el navegador web Chromium a pantalla completa.

Al comienzo, este sistema presenta una pantalla de inicio fija en la que no puede ejecutarse ninguna acción. Se mantiene a la espera de que el *Sistema de gestión* le envíe las configuraciones necesarias. La Figura 14 muestra la pantalla de inicio del simulador.



*Figura 14. Pantalla de inicio del simulador*

Luego de recibir dichas configuraciones, da comienzo inmediatamente a una nueva simulación, como puede verse en la Figura 15. En esta, se distinguen dos entidades:

- **Cañón:** este recurso es controlado por el operador del cañón. Lo que se ve a través de su mira es lo que se visualiza en la pantalla del simulador. Puede rotar libremente

de forma horizontal como también hacia arriba y hacia abajo. Tiene la capacidad de disparar y de activar o desactivar la función de predicción. Dos indicadores en la pantalla muestran el estado de estas dos acciones en todo momento.

- **Aviones:** estos sujetos aparecen y se mueven en la simulación de acuerdo a la configuración recibida. Se desplazan dentro del rango de alcance del cañón y, si no son impactados, desaparecen luego de salir del mismo. Por el contrario, si son derribados, caen hacia el suelo y desaparecen transcurridos unos segundos.



*Figura 15. Simulador en funcionamiento*

La simulación finaliza cuando todos los aviones configurados han desaparecido, ya sea por haber sido derribados o por haber salido del rango de alcance del cañón. En ese momento, el sistema regresa a la pantalla de inicio a la espera de nuevas configuraciones.

### **5.6.2.2 Gestor de entradas**

El *Gestor de entradas* es un servidor WebSocket, desarrollado con Node.js, que se encarga de leer las entradas físicas del control del cañón, las cuales son recibidas a través de puertos serie, y enviarlas al *Simulador* para que sean representadas en la simulación. Las entradas que maneja son: la rotación horizontal, la rotación vertical, el disparo y la

predicción. Estas son enviadas al *Simulador* en un objeto en formato JSON, el cual posee los siguientes atributos:

- **rotation:** es un objeto que representa las rotaciones del cañón. Sus atributos son:
  - **x:** es la rotación en el eje x (rotación vertical) en grados.
  - **y:** es la rotación en el eje y (rotación horizontal) en grados.
- **shooting:** es un valor booleano que indica si se está presionando el pedal de disparo.
- **predictor:** es un valor booleano que indica si está activada la función de predicción.

### 5.6.3 Servidor

El *Servidor* ha sido desarrollado enteramente con Node.js, un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome [56].

#### 5.6.3.1 Gestor de comunicación

El *Gestor de comunicación* es un servidor WebSocket a través del cual el *Simulador* y el *Controlador de la simulación* intercambian mensajes preestablecidos en formato JSON. Todos poseen un atributo *type* que permite identificar el tipo de mensaje enviado. Los tipos de mensaje y sus atributos son los siguientes:

- **role:** es enviado tanto por el *Simulador* como por el *Controlador de la simulación* al *Servidor* al momento de conectarse. Sus atributos son:
  - **role:** es una cadena de caracteres que indica el rol dentro de la comunicación. Sus posibles valores son: “*simulator*” para anunciarse como *Simulador* y “*manager*” para anunciarse como *Controlador de la simulación*.
- **state:** es enviado por el *Servidor* al *Controlador de la simulación* luego de que este le haya enviado el mensaje de tipo *role*. Sus atributos son:
  - **state:** es un valor booleano que indica si el *Simulador* está disponible para ser utilizado (*true*) o si ya está siendo utilizado por otra instancia de *Controlador de la simulación* (*false*).

- **setup:** es enviado por el *Controlador de la simulación* al *Servidor* y reenviado por este al *Simulador*. Contiene las configuraciones para una nueva simulación. Sus atributos son:
  - **simulation\_id:** es el identificador único de la simulación en la base de datos. Este valor es cargado por el *Servidor*.
  - **targets:** es un arreglo cuyos elementos contienen las configuraciones de los aviones agregados a la simulación. Los atributos de cada elemento son:
    - **id:** es el identificador único del avión dentro de la simulación. Es distinto del que se le asigna en la base de datos.
    - **distance:** es el valor de la distancia de la trayectoria.
    - **angle:** es el valor del ángulo de la trayectoria.
    - **height:** es el valor de la altura de la trayectoria.
    - **speed:** es el valor de la velocidad de la trayectoria.
    - **time:** es el valor del parámetro opcional *tiempo*.
    - **eval:** es el valor del parámetro opcional *evaluar*.
- **start:** es enviado por el *Simulador* al *Servidor* y reenviado por este al *Controlador de la simulación*. Indica que ha comenzado una nueva simulación. Sus atributos son:
  - **simulation\_id:** es el identificador único de la simulación en la base de datos.
- **target\_in:** es enviado por el *Simulador* al *Servidor* y reenviado por este al *Controlador de la simulación*. Indica que un nuevo avión ha aparecido en la simulación. Sus atributos son:
  - **simulation\_id:** es el identificador único de la simulación en la base de datos.
  - **id:** es el identificador único del avión dentro de la simulación.
  - **time\_in:** es el instante de tiempo en el que ha aparecido el avión, medido en milisegundos desde el comienzo de la simulación.

- **target\_pos:** es enviado por el *Simulador* al *Servidor* y reenviado por este al *Controlador de la simulación*. Representa un cambio en la posición de un avión. Sus atributos son:
  - **simulation\_id:** es el identificador único de la simulación en la base de datos.
  - **id:** es el identificador único del avión dentro de la simulación.
  - **time:** es el instante de tiempo en el que ha ocurrido el cambio, medido en milisegundos desde el comienzo de la simulación.
  - **x:** es la posición en el eje x en metros.
  - **y:** es la posición en el eje y en metros.
  - **z:** es la posición en el eje z en metros.
- **target\_rot:** es enviado por el *Simulador* al *Servidor* y reenviado por este al *Controlador de la simulación*. Representa un cambio en la rotación de un avión. Sus atributos son:
  - **simulation\_id:** es el identificador único de la simulación en la base de datos.
  - **id:** es el identificador único del avión dentro de la simulación.
  - **time:** es el instante de tiempo en el que ha ocurrido el cambio, medido en milisegundos desde el comienzo de la simulación.
  - **x:** es la rotación en el eje x en grados.
  - **y:** es la rotación en el eje y en grados.
  - **z:** es la rotación en el eje z en grados.
- **target\_hit:** es enviado por el *Simulador* al *Servidor* y reenviado por este al *Controlador de la simulación*. Indica que un avión ha sido derribado. Sus atributos son:
  - **simulation\_id:** es el identificador único de la simulación en la base de datos.
  - **id:** es el identificador único del avión dentro de la simulación.

- **time\_hit:** es el instante de tiempo en el que el avión ha sido derribado, medido en milisegundos desde el comienzo de la simulación.
- **target\_out:** es enviado por el *Simulador* al *Servidor* y reenviado por este al *Controlador de la simulación*. Indica que un avión ha desaparecido de la simulación, ya sea por haber sido derribado o por haber salido del rango de alcance del cañón. Sus atributos son:
  - **simulation\_id:** es el identificador único de la simulación en la base de datos.
  - **id:** es el identificador único del avión dentro de la simulación.
  - **time\_out:** es el instante de tiempo en el que ha aparecido el avión, medido en milisegundos desde el comienzo de la simulación.
- **cannon:** es enviado por el *Simulador* al *Servidor* y reenviado por este al *Controlador de la simulación*. Representa un cambio en la rotación del cañón. Sus atributos son:
  - **simulation\_id:** es el identificador único de la simulación en la base de datos.
  - **time:** es el instante de tiempo en el que ha ocurrido el cambio, medido en milisegundos desde el comienzo de la simulación.
  - **rotation\_x:** es la rotación en el eje x (rotación vertical) en grados.
  - **rotation\_y:** es la rotación en el eje y (rotación horizontal) en grados.
- **shooting:** es enviado por el *Simulador* al *Servidor* y reenviado por este al *Controlador de la simulación*. Indica que se ha comenzado o se ha dejado de disparar. Sus atributos son:
  - **simulation\_id:** es el identificador único de la simulación en la base de datos.
  - **time:** es el instante de tiempo en el que ha ocurrido el evento, medido en milisegundos desde el comienzo de la simulación.
  - **state:** es un valor booleano que indica si se ha comenzado (*true*) o se ha dejado (*false*) de disparar.

- **predictor:** es enviado por el *Simulador* al *Servidor* y reenviado por este al *Controlador de la simulación*. Indica que se ha activado o se ha desactivado la función de predicción. Sus atributos son:
  - **simulation\_id:** es el identificador único de la simulación en la base de datos.
  - **time:** es el instante de tiempo en el que ha ocurrido el evento, medido en milisegundos desde el comienzo de la simulación.
  - **state:** es un valor booleano que indica si se ha activado (*true*) o se ha desactivado (*false*) la función de predicción.
- **shot:** es enviado por el *Simulador* al *Servidor* y reenviado por este al *Controlador de la simulación*. Representa el disparo de una bala. Sus atributos son:
  - **simulation\_id:** es el identificador único de la simulación en la base de datos.
  - **time:** es el instante de tiempo en el que se ha producido el disparo, medido en milisegundos desde el comienzo de la simulación.
- **end:** es enviado por el *Simulador* al *Servidor* y reenviado por este al *Controlador de la simulación*. Indica que ha finalizado la simulación. Sus atributos son:
  - **simulation\_id:** es el identificador único de la simulación en la base de datos.
  - **time:** es el instante de tiempo en el que ha finalizado la simulación, medido en milisegundos desde el comienzo de la misma.
- **results:** es enviado por *Servidor* al *Controlador de la simulación*. Indica que ya se han calculado y almacenado en la base de datos las analíticas correspondientes a la simulación. Sus atributos son:
  - **simulation\_id:** es el identificador único de la simulación en la base de datos.

### 5.6.3.2 Generador de analíticas

El *Generador de analíticas* es un servidor HTTP que provee el cálculo de las analíticas de una simulación solicitada. Las analíticas son enviadas en un arreglo en formato JSON, en el que los elementos representan a los aviones que han aparecido en la simulación y cada uno posee los siguientes atributos:

- **target\_id:** es el identificador único del avión en la base de datos.
- **respon\_in:** es el instante de tiempo en el que el avión ha ingresado al sector de responsabilidad del cañón, medido en milisegundos desde el comienzo de la simulación.
- **respon\_out:** es el instante de tiempo en el que el avión ha salido del sector de responsabilidad del cañón, medido en milisegundos desde el comienzo de la simulación.
- **tracing:** es un objeto que indica los cambios en el porcentaje de seguimiento a lo largo del tiempo. Sus atributos son:
  - **time:** es un arreglo con los instantes de tiempo en los que han ocurrido los cambios, medidos en milisegundos desde el comienzo de la simulación.
  - **val:** es un arreglo, paralelo al anterior, con los valores que ha tomado el porcentaje de seguimiento en cada cambio.
- **respon:** es un objeto que indica los cambios en el porcentaje de responsabilidad a lo largo del tiempo. Sus atributos son:
  - **time:** es un arreglo con los instantes de tiempo en los que han ocurrido los cambios, medidos en milisegundos desde el comienzo de la simulación.
  - **val:** es un arreglo, paralelo al anterior, con los valores que ha tomado el porcentaje de responsabilidad en cada cambio.
- **shooting:** es un objeto que indica cuándo se ha comenzado o se ha dejado de disparar. Sus atributos son:
  - **time\_on:** es un arreglo con los instantes de tiempo en los que se ha comenzado a disparar, medidos en milisegundos desde el comienzo de la simulación.
  - **time\_off:** es un arreglo, paralelo al anterior, con los instantes de tiempo en los que se ha dejado de disparar, medidos en milisegundos desde el comienzo de la simulación.
- **predictor:** es un objeto que indica cuándo se ha activado o se ha desactivado la función de predicción. Sus atributos son:

- **time\_on:** es un arreglo con los instantes de tiempo en los que se ha activado la función de predicción, medidos en milisegundos desde el comienzo de la simulación.
- **time\_off:** es un arreglo, paralelo al anterior, con los instantes de tiempo en los que se ha desactivado la función de predicción, medidos en milisegundos desde el comienzo de la simulación.
- **shots:** es un arreglo con los instantes de tiempo en los que se ha disparado una bala, medido en milisegundos desde el comienzo de la simulación.

### 5.6.3.3 Gestor de base de datos

El *Gestor de base de datos* es el módulo del simulador responsable de establecer la conexión con la *Base de datos* y es utilizado para insertar, actualizar, leer y eliminar los registros en los que se almacena la información de las simulaciones ejecutadas.

### 5.6.4 Sistema de gestión

El *Sistema de gestión* es un sitio web a través del cual los usuarios pueden interactuar con el sistema. Este sitio web puede ser accedido desde cualquier computadora que se encuentre conectada por red a la computadora del cañón.

El sistema ha sido desarrollado conjuntamente con PHP para la programación del lado del servidor, HTML5 para la maquetación de las páginas, JavaScript para la programación del lado del cliente y CSS3 para los estilos de las páginas. Además, se ha utilizado jQuery, una librería de código abierto escrita en JavaScript que facilita la manipulación los elementos HTML, el manejo de eventos, las animaciones y la comunicación asíncrona con el servidor [57]. También se ha empleado Bootstrap, una librería de código abierto que brinda plantillas escritas en HTML5 y CSS3 para el diseño de sitios web, así como complementos desarrollados en JavaScript [58].

Inicialmente, el sistema presenta una página en la que el usuario debe iniciar sesión con su nombre de usuario y su contraseña para poder acceder al sistema, como se observa en la Figura 16.

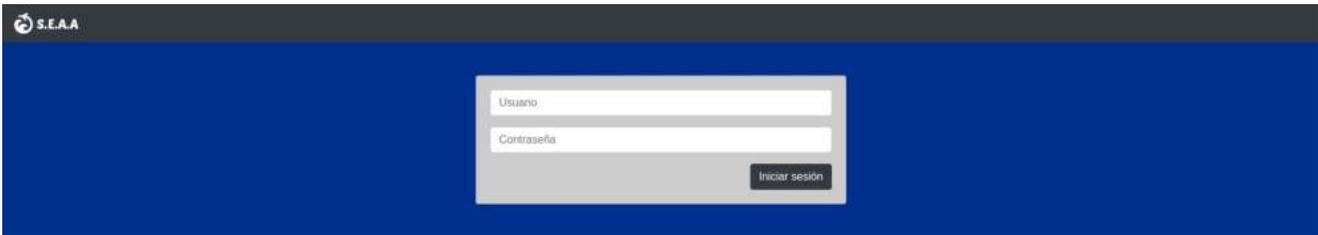


Figura 16. Pantalla de inicio de sesión

Luego, de acuerdo al rol del usuario, es redirigido a diferentes páginas. A los instructores se les presenta una página en la cual pueden ver una lista con los datos de todos los operadores registrados en el sistema, como la que se muestra en la Figura 17. Desde allí, pueden acceder al *Generador de reportes* y al *Controlador del simulador*.

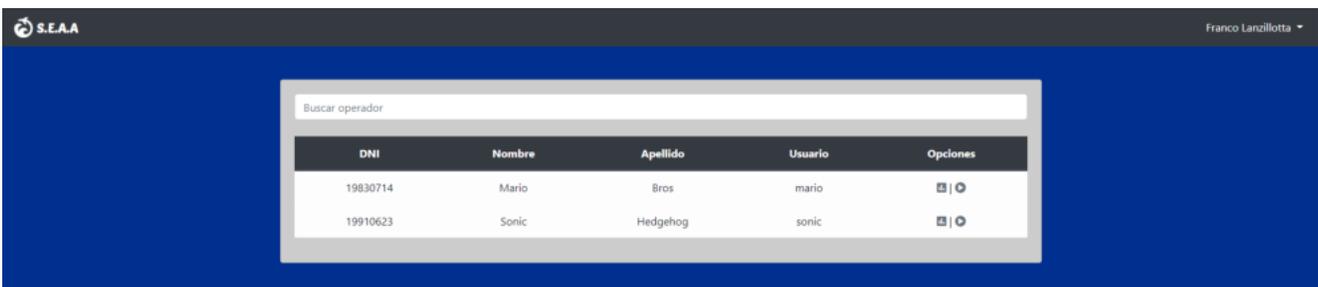


Figura 17. Listado de operadores

### 5.6.4.1 Controlador de la simulación

El *Controlador de la simulación* es una página a través de la cual el instructor puede configurar una nueva simulación y observar en tiempo real lo que ocurre en ella. Debido a que un operador no puede realizar simulaciones por sí mismo, si no que un instructor debe ser quien las controle, esta página solo es accesible para los instructores.

En primer lugar, al instructor se le presenta una página en la cual puede configurar la simulación, agregar aviones y establecer sus parámetros. La configuración de los aviones está acompañada con un gráfico con una vista cenital que permite comprender visualmente el efecto de los valores introducidos sobre la trayectoria del avión. Además, este gráfico muestra el campo visual del cañón al comienzo de la simulación y el sector de responsabilidad configurado. En la Figura 18 se muestra esta pantalla.

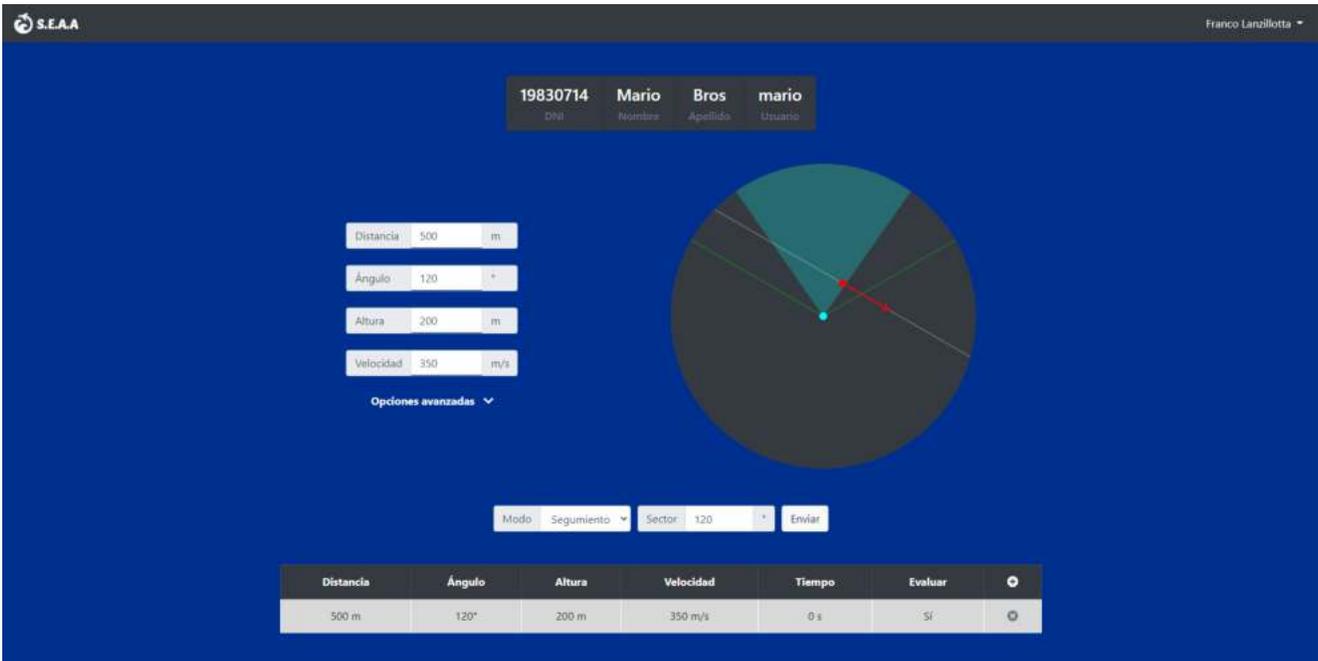


Figura 18. Pantalla de configuración de la simulación

Una vez iniciada la simulación, la página cambia para permitirle al instructor observar detalladamente todo lo que está ocurriendo en la simulación. En esta pantalla, puede verse a la izquierda lo que el operador está viendo en el *Simulador* en tiempo real. Mientras tanto, a la derecha se observa un gráfico con una vista cenital en el que pueden verse, en todo momento, las posiciones de los aviones, el campo visual del cañón y su sector de responsabilidad. Además, en la parte inferior hay dos indicadores que muestran el estado del disparo y de la función de predicción, junto con la cantidad de disparos efectuados. Debajo de estos, se presenta una tabla con los aviones agregados a la simulación, cuyas filas van cambiando de color de acuerdo a si el avión aparece o desaparece de la simulación o si es derribado. Esta pantalla puede verse en la Figura 19.

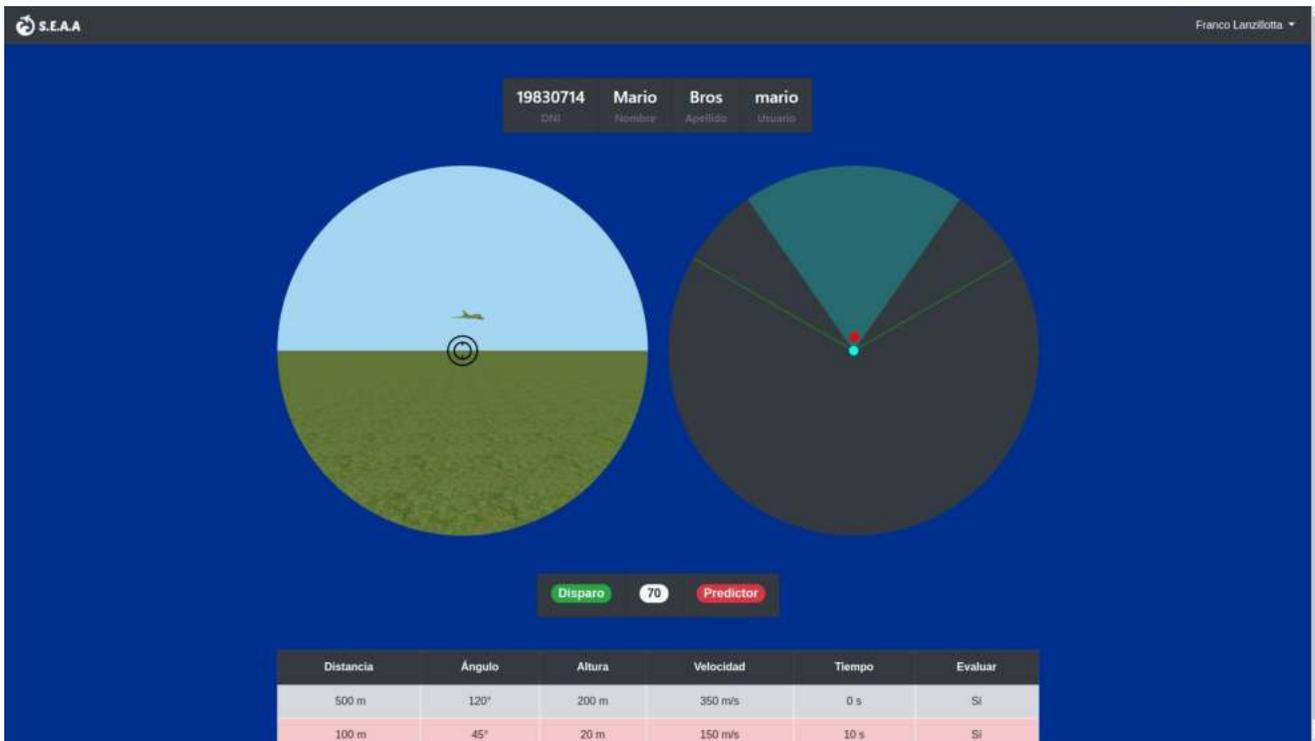


Figura 19. Pantalla de control de la simulación

### 5.6.4.2 Graficador de resultados

El *Graficador de resultados* es una página en la cual pueden visualizarse de forma detallada los valores de porcentaje de seguimiento y de responsabilidad obtenidos a lo largo del tiempo durante la simulación para cada uno de los aviones agregados a la misma. Además, en el gráfico se observa cuándo se ha disparado o se ha activado la función de predicción, cuándo se encuentra el avión dentro del sector de responsabilidad y en qué instante ha sido derribado. Debajo de este, se presenta una tabla con los aviones agregados a la simulación, a través de la cual se puede seleccionar el avión del cual se grafican sus valores. En esta tabla pueden verse los valores con los que se han configurado los aviones, acompañados de los valores de sus analíticas. Esta pantalla se muestra en la Figura 20.

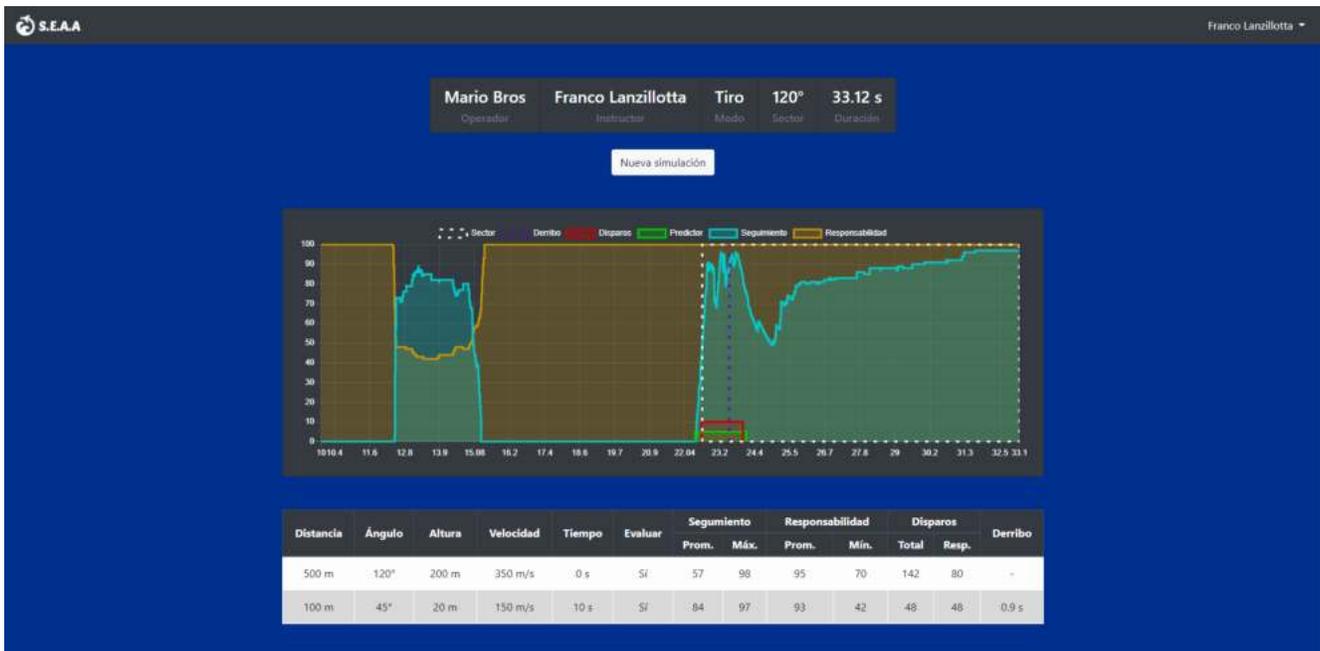


Figura 20. Graficador de resultados

### 5.6.5 Generador de reportes

El *Generador de reportes* es una página que permite la visualización de gráficos que muestran los valores de las analíticas obtenidas por un operador a lo largo de las simulaciones en el modo seleccionado, junto con el tiempo simulado en dicho modo. Desde allí, es posible acceder al *Graficador de resultados* para observar en detalle los resultados de cada una de estas simulaciones. Esta página es accesible para los instructores, los cuales pueden visualizar los reportes de cualquier operador. A su vez, es la página que se les muestra a los operadores cuando inician sesión con su usuario. Sin embargo, ellos solo pueden visualizar sus propios reportes. En la Figura 21 puede verse esta pantalla.

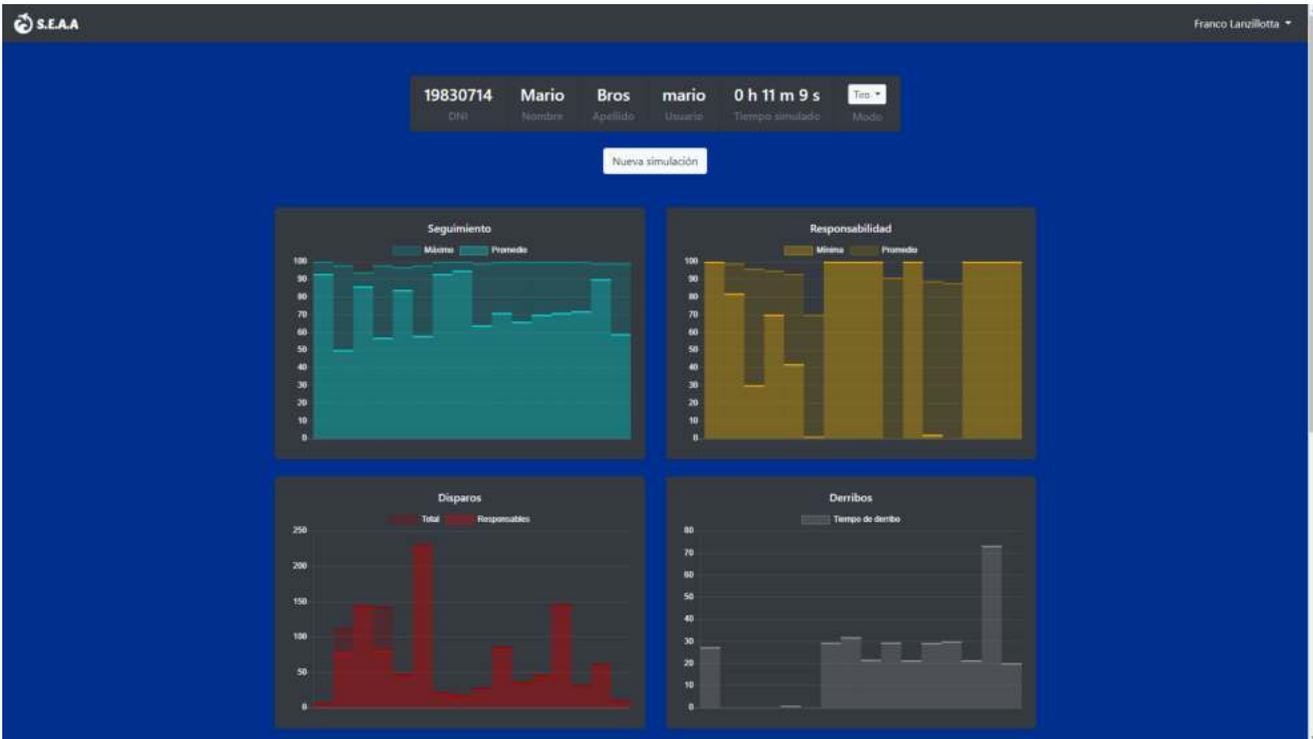
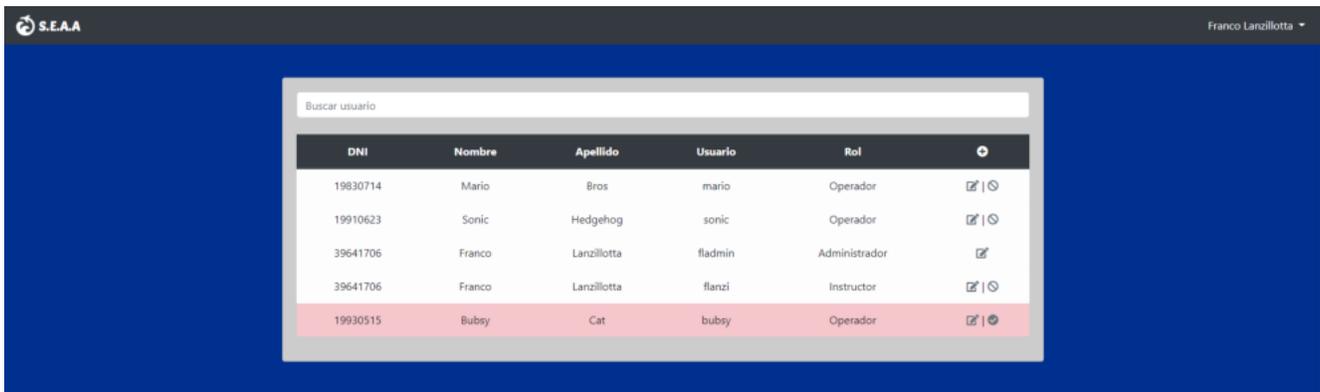


Figura 21. Generador de reportes

Tanto los gráficos del *Generador de reportes* como los del *Graficador de resultados* han sido realizados utilizando la librería Chart.js, la cual es muy fácil de usar, posee una buena y completa documentación, es sumamente personalizable y es de código abierto [59].

### 5.6.5.1 Gestor de usuarios

El *Gestor de usuarios* es una página en la cual los administradores pueden agregar, editar, activar, desactivar y eliminar usuarios. Cuando un usuario es desactivado, no es eliminado del sistema, solo se le inhabilita el acceso al mismo y, en el caso de los operadores, ya no son visibles sus reportes en el *Gestor de reportes* ni el detalle de sus simulaciones en el *Graficador de resultados*. Aún así, los usuarios pueden volverse a activar en cualquier momento. La eliminación solo puede ser realizada si el usuario no se encuentra asociado a ninguna simulación, ya sea como instructor o como operador, con el fin de evitar inconsistencias en los datos. En el caso de los administradores, el sistema garantiza que siempre exista al menos uno de ellos, por lo que si hay uno solo, este no puede ser eliminado. Esta pantalla se muestra en la Figura 22.



The screenshot shows a web application interface for user management. At the top left is the logo 'S.E.A.A' and at the top right is the name 'Franco Lanzillotta'. Below the header is a search bar labeled 'Buscar usuario'. The main content is a table with the following data:

DNI	Nombre	Apellido	Usuario	Rol	
19830714	Mario	Bros	mario	Operador	✎   ✖
19910623	Sonic	Hedgehog	sonic	Operador	✎   ✖
39641706	Franco	Lanzillotta	fladmin	Administrador	✎
39641706	Franco	Lanzillotta	flanzi	Instructor	✎   ✖
19930515	Bubby	Cat	bubby	Operador	✎   ✖

Figura 22. Gestor de usuarios

### 5.6.6 Base de datos

Dada la arquitectura del sistema, la *Base de datos* se encuentra alojada de forma local y el espacio de almacenamiento que posee es limitado. Por otro lado, el sistema ha sido pensado para ser utilizado por un pequeño número de usuarios, generalmente uno a la vez, por lo que el rendimiento y el manejo de concurrencia no resultaron factores críticos a la hora de seleccionar la tecnología a utilizar. Es por estas razones que se ha seleccionado una base de datos SQL para este sistema. Específicamente, se ha optado por el sistema de gestión SQLite, debido al reducido espacio de almacenamiento que requiere.

Luego de analizar las entidades relevantes del sistema y los datos asociados a cada una de ellas, se ha diseñado la *Base de datos*, la cual ha quedado definida como se muestra en la Figura 23.

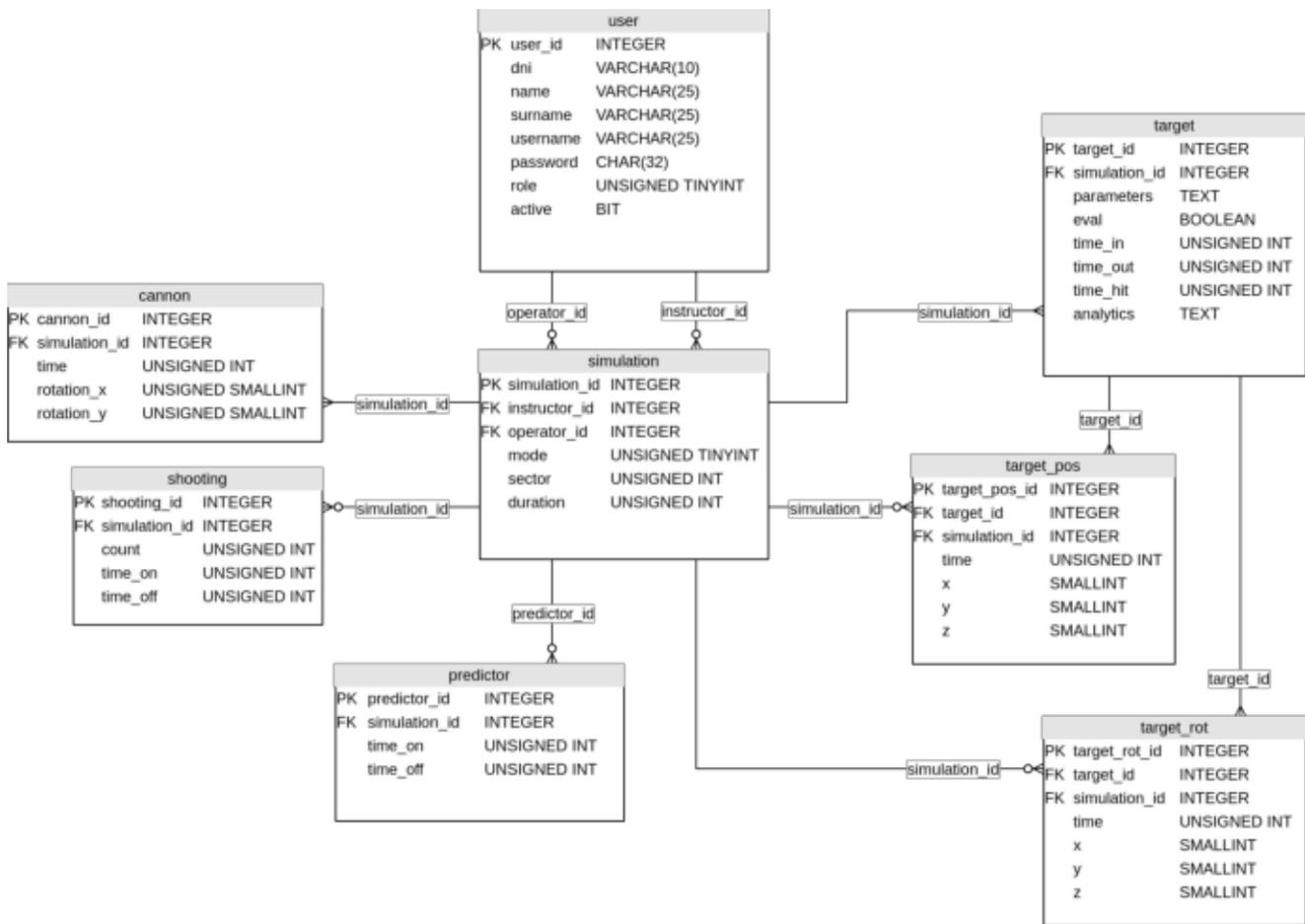


Figura 23. Diseño de la base de datos

- **Usuario (user):** es toda persona física que interactúa con el simulador. Posee un nombre de usuario y contraseña a fin de poder iniciar sesión en el sistema. Su rol puede ser: instructor, operador o administrador. Este último es quien se encarga de crear, editar, eliminar, activar o desactivar a los usuarios registrados en el sistema.
- **Simulación (simulation):** representa una sesión de simulación dentro del sistema. Tiene asociados un operador, que es quien realiza la simulación, y un instructor, que es quien supervisa y evalúa al operador.
- **Avión (target):** registra cada uno de los aviones que participan en cada simulación. Posee los tiempo de ingreso y egreso en la simulación así como el tiempo en el que fue derribado. Los parámetros y las analíticas se almacenan en formato JSON para

permitir que en un futuro puedan cambiarse sin alterar la estructura de la base de datos.

- **Posición del avión (target\_pos):** representa un cambio en la posición del avión dentro de la simulación en cualquiera de los tres ejes (x, y, z), junto con el tiempo en el que ocurrió.
- **Rotación del avión (target\_pos):** representa un cambio en la rotación del avión dentro de la simulación en cualquiera de los tres ejes (x, y, z), junto con el tiempo en el que ocurrió.
- **Rotación del cañón (cannon):** registra un cambio en la rotación del cañón, tanto horizontal como verticalmente, además del tiempo en el que ocurrió.
- **Disparo (shooting):** cada vez que se acciona el disparo, se almacena en qué momento comenzó, cuándo finalizó u cuántas municiones fueron disparadas.
- **Preditor (predictor):** cada vez que se activa la función de predicción, se almacena en qué momento comenzó y cuándo finalizó.

### 5.6.7 Transmisión de la pantalla del Simulador

Una de las características solicitadas por el demandante fue que el instructor pueda visualizar en tiempo real en el *Controlador de la simulación* lo que el operador está viendo en la pantalla del *Simulador*. Para llevar a cabo su desarrollo, se propusieron y evaluaron dos alternativas:

- **Transmisión de imagen:** el *Simulador* captura la imagen de la pantalla y la transmite al *Controlador de la simulación* para que sea mostrada en su interfaz gráfica.
- **Transmisión de estados:** el *Controlador de la simulación* utiliza los cambios de estado en la simulación que son transmitidos por el *Simulador* para generar una réplica de su pantalla a partir de los mismos.

#### 5.6.7.1 Transmisión de imagen

La primera alternativa probada ha sido la *Transmisión de imagen*. En esta propuesta, el *Simulador* captura su pantalla a una tasa fija. La imagen obtenida es escalada para reducir

su tamaño y facilitar su transmisión. Luego, es enviada a través de un WebSocket al *Controlador de la simulación*, el cual simplemente se encarga de representar la imagen recibida en su interfaz gráfica. Con el objetivo de reducir la cantidad de datos transmitidos, el *Simulador* no envía la cabecera de la imagen, ya que debido a que el tamaño de las imágenes es fijo, sus cabeceras son siempre las mismas. Al recibir la imagen, el *Controlador de la simulación* es quien se encarga de agregar la cabecera justo antes de mostrarla.

Esta propuesta permite variar dos valores propios de la transmisión: la tasa de captura de las imágenes y el tamaño de las mismas. El ajuste de estas variables establece una relación entre el rendimiento de la transmisión y la calidad de la visualización. Mientras menores sean la tasa de captura y el tamaño de la imagen, se requerirá un menor costo computacional y la cantidad de datos transmitidos será menor. Por el contrario, aumentar estos valores implica una mejora en la calidad de la visualización resultante en el *Controlador de la simulación*. En la Figura 24 se muestra un diagrama de esta alternativa.

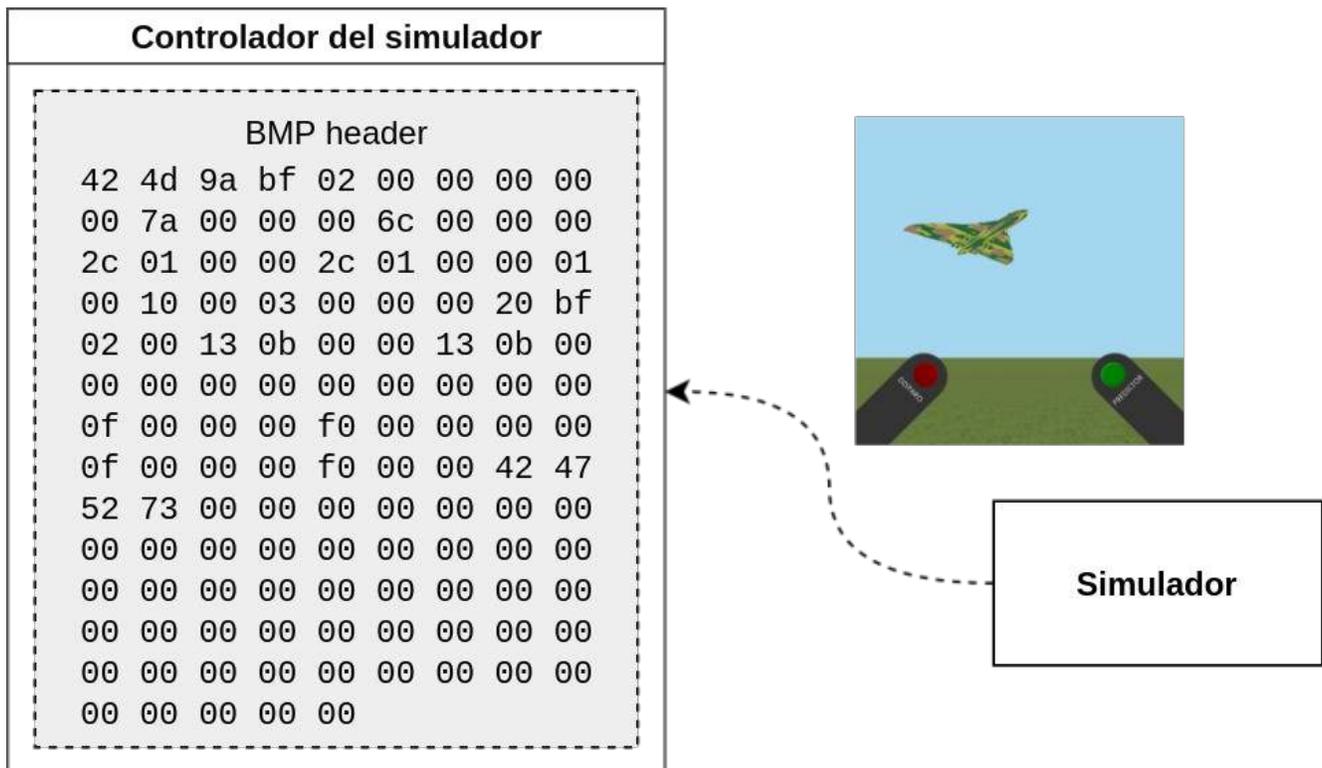


Figura 24. Transmisión de la pantalla del simulador - Transmisión de imagen

### **Ventajas**

- Se representa de forma fiel lo que se ve en la pantalla del *Simulador*.
- La lógica de la transmisión se mantiene en el *Simulador*.

### **Desventajas**

- Requiere un costo computacional elevado en el *Simulador*.
- Se transmite una gran cantidad de datos.
- La calidad de la visualización se ve comprometida.

#### **5.6.7.2 Transmisión de estados**

La otra alternativa probada ha sido la *Transmisión de estados*. Esta propuesta no requiere que se transmitan nuevos datos, ya que se utilizan los cambios de estados que ya son transmitidos para otros fines. Por lo tanto, toda la lógica está en el *Controlador de la simulación*. Este módulo recrea, a partir de los cambios de estado, lo que se está viendo en la pantalla del *Simulador*. Para llevar a cabo esta tarea, se ha seleccionado Three.js, una librería escrita en JavaScript que permite crear y mostrar gráficos 3D animados en navegadores web [60]. Esta alternativa puede verse graficada en la Figura 25.

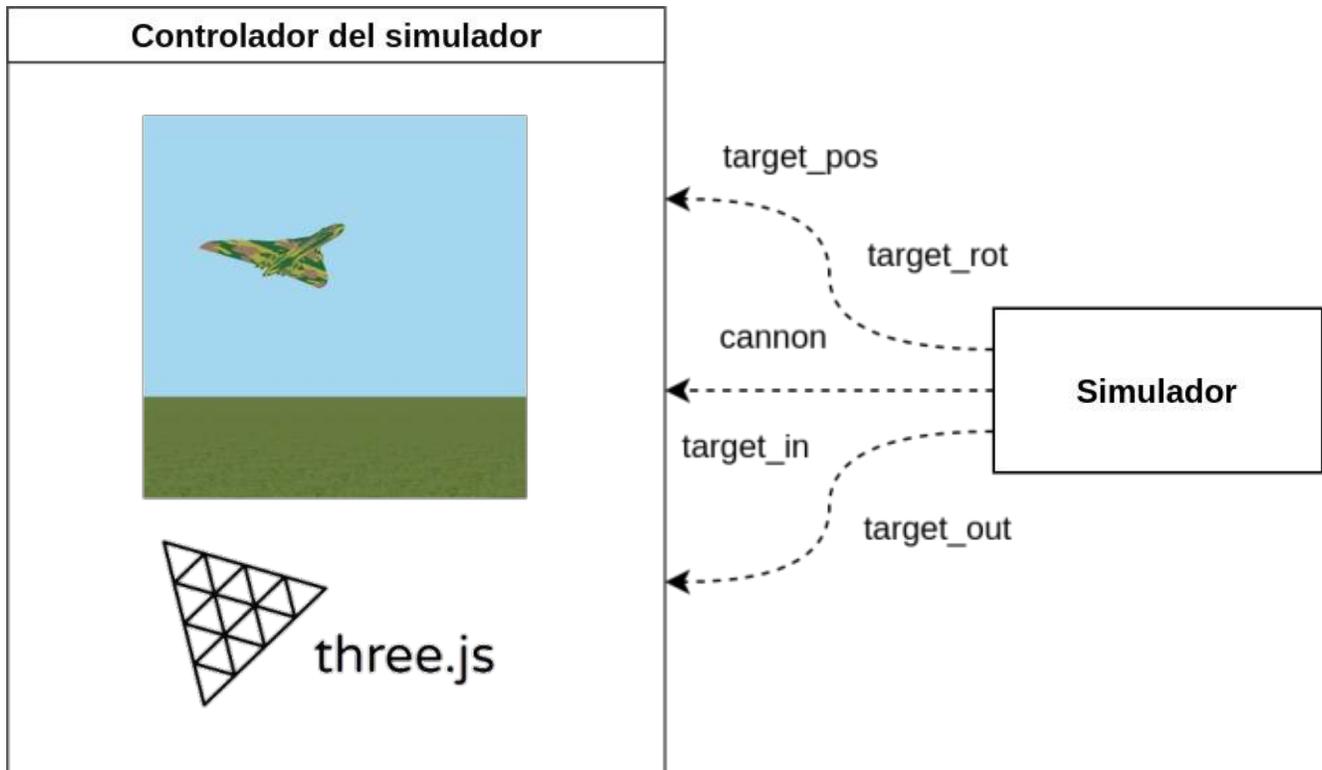


Figura 25. Transmisión de la pantalla del simulador - Transmisión de estados

### Ventajas

- No requiere que se transmitan más datos.
- El *Simulador* no necesita realizar nuevas tareas.
- La calidad de la visualización no se ve comprometida.

### Desventajas

- La lógica se traslada al *Controlador de la simulación*.
- No puede garantizarse que la visualización represente exactamente lo que está ocurriendo en la simulación.

### 5.6.7.3 Selección de la alternativa

Ambas alternativas propuestas fueron desarrolladas y probadas dentro del sistema, utilizando los dispositivos seleccionados. Luego de estas pruebas, se comprobó que el costo

computacional elevado de la *Transmisión de imagen* provoca como consecuencia que el *Simulador* no pueda ejecutarse correctamente, haciendo imposible su uso. La *Transmisión de estados*, por su parte, ha demostrado funcionar correctamente. Si bien puede que la representación proporcionada por esta alternativa no sea exacta, es más que suficiente para permitir que los instructores puedan visualizar y comprender lo que está ocurriendo en la simulación. Por lo tanto, de acuerdo con lo expuesto y las ventajas mencionadas, se ha decidido seleccionar la alternativa de la *Transmisión de estados* para transmitir en tiempo real la pantalla del *Simulador* al *Controlador de la simulación*.

## Capítulo 6: Conclusiones y trabajos futuros

Los Simuladores de Entrenamiento se están consolidando como una herramienta efectiva para brindar capacitación en ambientes y situaciones que impliquen un gran riesgo o conlleven costos altos. Sin embargo, la inclusión de Analíticas de Aprendizaje en estos sistemas es una temática incipiente que aún se encuentra en desarrollo. En este contexto, se propuso llevar a cabo la puesta en valor de un Simulador de Entrenamiento, mediante la incorporación de experiencia inmersiva y el uso de Analíticas de Aprendizaje para asistir en la evaluación de los aprendices.

En el marco del MPDSG, se ha realizado la elicitación del Simulador de Entrenamiento y la especificación obtenida se ha sometido en cada iteración al juicio de los expertos para validarla y mejorarla. A través de este proceso, se han logrado confeccionar escenarios de Leite que permitieron adaptar el simulador a las necesidades de sus usuarios. Además, se han podido definir de forma clara y precisa las habilidades que se desean entrenar y las Analíticas de Aprendizaje con las que se quiere evaluar el cumplimiento de las mismas.

A partir de los requerimientos relevados, se han definido la arquitectura, las componentes y el flujo de datos del sistema, en función de las necesidades manifestadas por los expertos. Luego, se realizó un proceso de selección de los motores de videojuegos de propósito general disponibles en el mercado, con lo que se ha determinado el más apto para el desarrollo del Simulador de Entrenamiento en base a las necesidades del diseño propuesto.

Seguidamente, se ha desarrollado el sistema, implementando cada una de las componentes definidas en la fase de diseño y la comunicación necesaria entre las mismas. Se han utilizado, en su mayoría, tecnologías conocidas por el autor para así reducir los tiempos de desarrollo.

Se ha conseguido como resultado un sistema que debería ser fácil de mantener y adaptar de acuerdo a las nuevas necesidades que puedan surgir a medida que sea utilizado. Esto puede verse ejemplificado en el bajo acoplamiento de sus componentes, lo que permitiría que una de ellas pueda modificarse sin impactar de forma significativa en el resto.

De igual modo, aquellos datos que pueden ser susceptibles de requerir cambios en un futuro (como los parámetros de configuración de la simulación o las analíticas resultantes)

han sido estructurados con la intención de que estos cambios puedan ser efectuados fácilmente. Además, los datos provenientes de la simulación son independientes del funcionamiento del sistema, dado que representan los estados de los actores y recursos de la simulación en cada instante de tiempo. Esto debería posibilitar que el simulador pueda ser modificado sin necesidad de efectuar cambios en el resto del sistema.

Si bien el desarrollo ha sido pensado en base al hardware disponible a la fecha, se ha concebido una solución con una baja dependencia del mismo. El simulador debería poder ser portado fácilmente si en un futuro se requiriera, ya que se han empleado tecnologías compatibles con la mayoría de las plataformas disponibles en el mercado. Adicionalmente, el sistema de gestión puede ser accedido desde cualquier computadora a través de cualquier navegador web moderno, sin necesidad de instalar ningún tipo de software adicional.

El uso de metodologías ha contribuido a la calidad de la solución obtenida. A través del proceso iterativo del MPDSG, se ha mantenido una comunicación constante con los expertos, quienes han validado los resultados que se han ido obteniendo a lo largo de las iteraciones. Esto ha posibilitado corregir las desviaciones de forma temprana y que el sistema pueda adaptarse de mejor forma a los requerimientos de los distintos usuarios. Por otro lado, el marco de trabajo GLEANER ha brindado lineamientos para la implementación de Analíticas de Aprendizaje en el sistema, de manera que las mismas puedan llegar a ser de utilidad a la hora de evaluar objetivamente a los aprendices.

Este trabajo también representa un aporte al conocimiento de las áreas involucradas. Su desarrollo colabora en la incorporación de nuevas tecnologías en la formación profesional, específicamente en el uso de videojuegos como herramientas para la adquisición de conocimientos y habilidades. Por otro lado, contribuye en la definición de un marco de referencia sobre la incorporación de Analíticas de Aprendizaje en Simuladores de Entrenamiento, una temática de interés en la actualidad que se encuentra en desarrollo.

Finalmente, este trabajo ha significado para el autor un gran aporte en la integración de los conocimientos adquiridos a lo largo de la carrera, por lo que ha contribuido de forma significativa a su formación profesional. Ha sido su primera experiencia en la realización de un proyecto de tal magnitud. Más allá del aporte a los conocimientos técnicos, ha brindado valiosas lecciones sobre la importancia de la comunicación constante con los clientes, de la

aplicación de principios y buenas prácticas en el diseño de un sistema, de la correcta estimación de tiempos, entre muchas otras.

## 6.1 Limitaciones y trabajos futuros

Este trabajo ha finalizado con la puesta en marcha del sistema, lo que concluye la fase de producción definida en el MPDSG. El siguiente paso debería implicar la realización de pruebas exhaustivas del sistema en sesiones de entrenamiento con los aprendices. De este modo, se podrá obtener la retroalimentación necesaria para evaluar la eficacia del sistema como herramienta de enseñanza y aprendizaje, el cual podrá ajustarse y mejorarse de acuerdo a las necesidades de sus usuarios.

A lo largo del proceso de desarrollo, los expertos han ido sugiriendo posibles mejoras a implementar en una futura etapa, entre las que se incluyen:

- Brindar la posibilidad de seleccionar la silueta de los aviones simulados, dado que permitiría evaluar la habilidad del operador en el reconocimiento de aviones amigos y enemigos.
- Incorporar dificultades propias de la realidad, tales como distintos escenarios con obstáculos para la visión y la posibilidad de introducir variaciones en el clima, como la lluvia o la posición del sol.
- Agregar efectos visuales y de sonido, como bombas, disparos o gritos, para generar en el operador sensaciones de estrés y nerviosismo más cercanas a las que se producen en situaciones reales.
- Integrar este sistema junto con otros simuladores que se utilizan para la formación de otras habilidades, con lo cual se podría brindar un entrenamiento cooperativo que involucren simultáneamente diversas actividades.

La incorporación de las recomendaciones de los expertos permitiría mejorar aún más la experiencia inmersiva y la sensación de realismo del simulador. Asimismo, brindaría la posibilidad de entrenar y evaluar nuevas habilidades, las cuales complementarían a las que se han previsto en esta etapa.

## Bibliografía

- [1] Bunk, G. P. (1994). La transmisión de las competencias en la formación y perfeccionamiento profesionales de la RFA. *Revista europea de formación profesional*, (1), pp. 8–14.
- [2] Ramón del Rey, E. (2019). Los juegos de simulación empresarial: aplicación practica en el contexto de la Formación Profesional. Trabajo Fin de Máster, Escuela Internacional de Máster, Universidad de Almería.
- [3] Valderrama Ramos, J. A. (2012). Los videojuegos: conectar alumnos para aprender. *Sinéctica*, (39), pp. 1–15.
- [4] Luppá, N. y Borst, T. (2013). *Story and simulations for serious games: Tales from the trenches*. United States: Focal Press, Taylor & Francis Group.
- [5] Marsh, T., Wong, W. L., Carriazo, E., Nocera, L., Yang, K., Varma, A., Yoon H., Huang Y-L, Kyriakakis, C. y Shahabi, C. (2005). User Experiences and Lessons Learned from Developing and Implementing an Immersive Game for the Science Classroom. *Proceedings of HCI International 2005*, Las Vegas, Nevada, USA.
- [6] Darfur is Dying (sin fecha). Wikipedia. Recuperado el 29 de mayo de 2020 de [https://en.wikipedia.org/wiki/Darfur\\_is\\_Dying](https://en.wikipedia.org/wiki/Darfur_is_Dying).
- [7] Kühn, F. D. (2019). Juegos Serios y Analíticas de Aprendizaje: Implementación en el entorno educativo. Trabajo Final de Grado, Facultad de Ingeniería, Universidad Nacional de Mar del Plata.
- [8] Lechtaler, A. C., Blanc, C. F., Carden, M. L., Köhler, A., Polak, A. G. y Señorino, J. M. (2015). Simulación Inmersiva con Realidad Aumentada. XVII Workshop de Investigadores en Ciencias de la Computación (WICC), Salta, Salta, Argentina.
- [9] Advenio Software (sin fecha). Vox Maris - Simulador GMDSS. Recuperado el 4 de junio de 2020 de <https://www.advenio.com.ar/es/voymaris>.
- [10] Rhienmora, P., Gajananan, K., Haddawy, P., Dailey, M. N. y Suebnukarn, S. (2010). Augmented Reality Haptics System for Dental Surgical Skills Training. *Proceedings*

of the 17th ACM Symposium on Virtual Reality Software and Technology (VRST), pp. 97–98.

- [11] Johnson, L., Adams Becker, S., Cummins, M., Estrada, V., Freeman, A. y Hall, C. (2016). NMC Horizon Report: 2016 Higher Education Edition. Austin, Texas: The New Media Consortium.
- [12] Serrano-Laguna, A. (2017). Mejorando la evaluación de juegos serios mediante el uso de analíticas de aprendizaje. Tesis Doctoral, Facultad de Informática, Universidad Complutense de Madrid.
- [13] Michael, D. y Chen, S. (2006). Serious Games: Games That Educate, Train and Inform. Boston: Thomson.
- [14] Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer*, 38(9), pp. 25–32.
- [15] Guenaga, M., Eguíluz, A., Rayón, A., y Quevedo, E. (2015). Un juego Serio para Desarrollar y Evaluar la Competencia de Trabajo en Equipo. *IE Comunicaciones: Revista Iberoamericana de Informática Educativa*, (21), pp. 3–11.
- [16] Prensky, M. (2003). Digital game-based learning. *Computers in Entertainment (CIE)*, 1(1), pp. 21–21.
- [17] Csikszentmihalyi, M. (1998). *Creatividad: El flujo y la psicología del descubrimiento y la invención*. Paidós, Barcelona.
- [18] PeaceMaker (sin fecha). Wikipedia. Recuperado el 28 de abril de 2020 de <https://en.wikipedia.org/wiki/PeaceMaker>
- [19] Laboratorio de Investigación en Nuevas Tecnologías Informáticas (2014). Más allá de las palabras: Un juego social para revalorizar las culturas originarias. Recuperado el 28 de abril de 2020 de <http://raices.linti.unlp.edu.ar/proyecto>
- [20] Gamelearn (sin fecha). Pacific: Serious Game de Liderazgo y Gestión de Equipos. Recuperado el 28 de abril de 2020 de <https://www.game-learn.com/game-based-learning-formacion-corporativa/serious-game-de-liderazgo-y-gestion-de-equipos/>

- [21] Vidal-Gomel, C. y Fauquet-Alekhine, P. (2016). Reflections and theoretical contributions regarding trainers' practice and simulation. *Simulation training: Fundamentals and applications – Improving professional practice through simulation training*, Springer, Cham, pp. 1–29.
- [22] Drews, F. A. y Bakdash, J. Z. (2013). Simulation Training in Health Care. *Reviews of Human Factors and Ergonomics*, 8(1), pp. 191–234.
- [23] Vázquez-Mata, G. y Guillamer Lloveras, A. (2009). El entrenamiento basado en la simulación como innovación imprescindible en la formación médica. *Educación Médica*, 12(3), pp. 149–145.
- [24] Camarasa, D. A. y Bianchi, O. M. (2012). Desarrollo de Software de Simulación Inmersiva para Fracciones Heterogéneas. XIV Workshop de Investigadores en Ciencias de la Computación (WICC), Posadas, Misiones, Argentina.
- [25] Luciano, C., Banerjee, P. y DeFanti, T. (2009). Haptics-based virtual reality periodontal training simulator. *Virtual Reality*, 13(2), pp. 69–85.
- [26] Microsoft Flight Simulator (sin fecha). Wikipedia. Recuperado el 28 de abril de 2020 de [https://es.wikipedia.org/wiki/Microsoft\\_Flight\\_Simulator](https://es.wikipedia.org/wiki/Microsoft_Flight_Simulator)
- [27] VICE — DAS (sin fecha). Recuperado el 28 de abril de 2020 de <https://www.d-a-s.com/vice>
- [28] Centro de Simulación y Juegos de Guerra (CSJG) (sin fecha). Recuperado el 28 de abril de 2020 de <https://www.argentina.gob.ar/fuerzaaerea/direccion-general-de-investigacion-y-desarrollo/centro-de-simulacion-y-juegos-de-guerra-csjg>
- [29] Ferguson, R. (2012). Learning analytics: drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, 4(5/6), pp. 304–317.
- [30] Ferguson, R. y Shum, S. B. (2012). Social learning analytics: Five Approaches. LAK '12: Proceedings of the 2nd International Conference on Learning Analytics and Knowledge, pp. 23–33.
- [31] Hershkovitz, A., Knight, S., Dawson, S., Jovanovic, J. y Gasevic, D. (2016). About “learning” and “analytics”. *Journal of Learning Analytics*, 3(2), pp. 1–5.

- [32] Drachen, A., Seif El-Nasr, M. y Canossa, A. (2013). *Game Analytics: Maximizing the Value of Player Data*. Springer.
- [33] Alonso Fernández, C. (2016). *Gaming Learning Analytics for Serious Games*. Trabajo de Fin de Grado, Facultad de Informática, Universidad Complutense de Madrid.
- [34] Van Eck, R. (2006). Digital Game-Based Learning: It's Not Just the Digital Natives Who Are Restless. *EDUCAUSE Review*, 41(2), pp. 16–31.
- [35] Serrano-Laguna, A., Torrente, J., Moreno-Ger, P. y Fernández-Manjón, B. (2012). Tracing a Little for Big Improvements: Application of Learning Analytics and Videogames for Student Assessment. *Procedia Computer Science*, 15, pp. 203–209.
- [36] Kitchenham, B. (2004). *Procedures for Performing Systematic Reviews*. Joint Technical Report, Keele University TR/SE-0401 and NICTA 0400011T.1.
- [37] Gonçalves, A. L., Carlos, L. M., da Silva, J. B. y Alves, G. R. (2018). Personalized Student Assessment based on Learning Analytics and Recommender Systems. 3rd International Conference of the Portuguese Society for Engineering Education (CISPÉE).
- [38] Tibola, L. R. (2014). A proposal of engineering education architecture: Improve engineer's competencies through practice labs and 3D virtual worlds. 12th IEEE International Conference on Industrial Informatics (INDIN).
- [39] Yin, M. S., Haddawy, P., Suebnukarn, S. y Rhiemora, P. (2018). Automated outcome scoring in a virtual reality simulator for endodontic surgery. *Computer Methods and Programs in Biomedicine*, 153, p. 53-59.
- [40] Doderó, J. M., González-Conejero, E. J., Gutiérrez-Herrera, G., Peinado, S., Tocino, J. T. y Ruiz-Rube, I. (2017). Trade-off between interoperability and data collection performance when designing an architecture for learning analytics. *Future Generation Computer Systems*, 68, p. 31-37.
- [41] Xanthopoulos, S. y Xinogalos, S. (2018). Opportunities and challenges of mobile location-based games in education: Exploring the integration of authoring and

- analytics tools. 2018 IEEE Global Engineering Education Conference (EDUCON), p. 1797-1805.
- [42] Andrade, A. (2017). Understanding student learning trajectories using multimodal learning analytics within an embodied-interaction learning environment. LAK '17: Proceedings of the Seventh International Learning Analytics & Knowledge Conference, p. 70-79.
- [43] Evans, F., Spinelli, A., Zapirain, E., Massa, S. M. y Soriano, F. (2016). Proceso de desarrollo de Serious Games. Diseño centrado en el usuario, jugabilidad e inmersión. III Congreso Argentino de Ingeniería (CADI) – IX Congreso Argentino de Enseñanza de la Ingeniería (CAEDI), Resistencia, Chaco, Argentina.
- [44] Baalsrud Hauge, J., Berta, R., Fiucci, G. y Fernández Manjón, B. (2014). Implications of learning analytics for serious game design. Proceedings of the 14th International Conference on Advanced Learning Technologies (ICALT), Athens, Greece, pp. 230–232.
- [45] Massa, S. M. (2013). Objetos de Aprendizaje: Metodología de Desarrollo y Evaluación de Calidad. Tesis Doctoral, Facultad de Informática, UNLP, La Plata.
- [46] Clements, P. y Northrop, L. M. (2002). Software Product Lines: Practices and Patterns. Addison-Wesley.
- [47] Hassan Montero, Y. y Ortega Santamaría, S. (2009). Informe APEI sobre usabilidad. Informe APEI 3.
- [48] Rogers, S. (2010). Level UP! The Guide to Great Video Games Design. John Wiley & Sons Inc., Estados Unidos.
- [49] Spinelli, A. T. y Massa, S. M. (2018). Elicitación en Serious Game. Congreso Bial IEEE ARGENCON 2018, San Miguel de Tucumán, Tucumán, Argentina.
- [50] Leite, J. C. S. P., Haddad, G. D. S., Doorn, J. H. y Kaplan, G. N. (2000). A Scenario Construction Process. Requirement Engineering, 5(1), pp. 38–61.

- [51] Spinelli, A. T., Massa, S. M., Rico, C. y Kühn, F. (2018) Diseño de Serious Games. Requerimientos del Juego – Competencias y Habilidades. XX Encuentro Internacional VirtualEduca, Buenos Aires, Argentina.
- [52] Westera, W., Nadolski, R. y Hummel, H. (2014). Serious Gaming Analytics: What Students Log Files Tell Us about Gaming and Learning. International Journal of Serious Games, 1(2), pp. 35–50.
- [53] Serrano-Laguna, A. (2012). GLAS: framework to improve assesment in educational videogames. Tesis de Maestría, Facultad de Informática, Universidad Complutense de Madrid.
- [54] Evans, F., Mileta, S. E. y Petrillo, J. D. (2016). Proceso de selección de tecnología para el desarrollo de Serious Games. II Jornadas Argentinas de Tecnología, Innovación y Creatividad (JATIC), Mar del Plata, Buenos Aires, Argentina.
- [55] Godot (sin fecha). Wikipedia. Recuperado el 12 de febrero de 2020 de <https://es.wikipedia.org/wiki/Godot>
- [56] Node.js (sin fecha). Recuperado el 13 de febrero de 2020 de <https://nodejs.org/>
- [57] jQuery (sin fecha). Recuperado el 29 de febrero de 2020 de <https://jquery.com/>
- [58] Bootstrap · The most popular HTML, CSS, and JS library in the world (sin fecha). Recuperado el 29 de febrero de 2020 de <https://getbootstrap.com/>
- [59] Chart.js | Open source HTML5 Charts for your website (sin fecha). Recuperado el 14 de febrero de 2020 de <https://www.chartjs.org/>
- [60] Three.js (sin fecha). Wikipedia. Recuperado el 29 de febrero de 2020 de <https://en.wikipedia.org/wiki/Three.js>

## **Anexo I – Protocolo de entrevistas a los expertos**

### **Expertos en simulación**

- 1) ¿El proyecto es nuevo o ya se cuentan con avances previos o prototipos realizados?
- 2) ¿Cuáles son las limitaciones que poseen los mecanismos actuales de entrenamiento?
- 3) ¿Qué cosas se pretenden mejorar respecto a los mecanismos actuales de entrenamiento?
- 4) ¿En qué consiste una sesión en el simulador?
- 5) ¿Cuáles son los eventos de la realidad que se pretenden simular?
- 6) ¿Se pretende simular un único blanco o varios simultáneamente?
- 7) ¿Qué parámetros se deberán configurar en la simulación?
- 8) ¿Quién será responsable de configurar esos parámetros?
- 9) ¿Qué debería visualizar el instructor durante y luego de la simulación?
- 10) ¿Se ha pensado en incorporar elementos físicos que favorezcan la sensación de realismo dentro de la simulación?

## Expertos en capacitación

- 1) ¿A qué grupo de personas estará destinado el simulador?
- 2) ¿La simulación se ha planteado para ser realizada individualmente o en grupo?
- 3) ¿Cuál considera que debería ser la duración de una sesión de simulación?
- 4) ¿Qué conocimientos mínimos debe tener el aprendiz para utilizar el simulador?
- 5) ¿Qué habilidades necesita poseer el aprendiz al momento de utilizar el simulador?
- 6) ¿Qué objetivos tiene el aprendizaje a través del simulador?
- 7) ¿Qué resultados relacionados con el desempeño del aprendiz considera relevantes visualizar durante la simulación?
- 8) ¿Podría resultar relevante visualizar las acciones del aprendiz durante la simulación?
- 9) ¿Cómo espera evaluar a través del simulador el cumplimiento de los objetivos de aprendizaje?
- 10) ¿Qué resultados de la simulación considera de interés a fin de evaluar el desempeño del aprendiz?
- 11) ¿Qué tareas se espera que el aprendiz realice a lo largo de una sesión de simulación?
- 12) ¿Cuáles considera que serían resultados ideales en el desempeño del aprendiz?
- 13) ¿Cuáles son los errores que cometen los aprendices durante el entrenamiento?
- 14) ¿El simulador será utilizado junto con materiales complementarios?
- 15) ¿Qué información relevante se le podría brindar al aprendiz luego de su instrucción con el objetivo de que pueda tener un seguimiento de su progreso?
- 16) ¿Qué elementos considera que contribuirán a mejorar la sensación de realismo dentro de la simulación?

## Expertos en física

- 1) ¿Cuáles son las características del cañón que resultan de interés a la hora de simularlo?
- 2) ¿Cuáles son los parámetros físicos que definen a un cañón?
- 3) ¿Qué controles posee el cañón? ¿Para qué se utiliza cada uno?
- 4) ¿Cómo se comporta el cañón durante su movimiento, inclusive al iniciarse y al detenerse?
- 5) ¿Cómo es la secuencia de acciones habitual a la hora de derribar una formación de aviones enemigos?
- 6) ¿Qué efecto produce sobre un avión el impacto de una bala del cañón utilizado en el simulador?
- 7) ¿Qué masa tiene la bala del cañón y a qué velocidad sale disparada del mismo?
- 8) ¿Cuáles son los parámetros físicos que definen a un avión?
- 9) ¿Cómo es el comportamiento de los aviones cuando se dirigen a zonas enemigas?
- 10) ¿En qué cantidades suelen movilizarse los aviones?
- 11) ¿Cómo son las posibles formaciones que describen los aviones?

## Anexo II – Definición de Escenarios

Tabla 5. Escenario de Leite – Configurar simulación

<b>Configurar simulación</b>	
<b>Objetivo</b>	Adaptar la simulación a las necesidades del entrenamiento
<b>Contexto</b>	El operador se encuentra ubicado en el cañón esperando a que comience la simulación
<b>Recursos</b>	Cañón
<b>Actores</b>	Instructor, Operador del cañón
<b>Set de Episodios</b>	<ul style="list-style-type: none"><li>• El instructor abre la interfaz de configuración de la simulación</li><li>• El instructor selecciona el sector de responsabilidad y el modo de simulación</li><li>• El instructor añade aviones a la simulación y configura sus parámetros</li><li>• El instructor da comienzo a la simulación</li></ul>
<b>Casos Alternativos</b>	<ul style="list-style-type: none"><li>• El instructor detiene el proceso sin dar comienzo a la simulación</li></ul>

*Tabla 6. Escenario de Leite – Buscar objetivo*

<b>Buscar objetivo</b>	
<b>Objetivo</b>	Encontrar al objetivo enemigo
<b>Contexto</b>	El cañón se encuentra en una determinada posición. Un avión enemigo se acerca volando al sector de responsabilidad del cañón
<b>Recursos</b>	Cañón
<b>Actores</b>	Operador del cañón, Avión
<b>Set de Episodios</b>	<ul style="list-style-type: none"> <li>• El operador desplaza al cañón en busca del objetivo</li> <li>• El operador visualiza al objetivo</li> <li>• El operador posiciona al objetivo en la mira</li> </ul>
<b>Casos Alternativos</b>	<ul style="list-style-type: none"> <li>• El operador no encuentra al objetivo</li> <li>• El operador no logra posicionar al objetivo en la mira</li> </ul>

*Tabla 7. Escenario de Leite – Seguir objetivo*

<b>Seguir objetivo</b>	
<b>Objetivo</b>	Realizar el seguimiento del objetivo enemigo
<b>Contexto</b>	El operador está apuntando al objetivo a través de la mira del cañón
<b>Recursos</b>	Cañón
<b>Actores</b>	Operador del cañón, Avión
<b>Set de Episodios</b>	<ul style="list-style-type: none"> <li>• El operador mueve el cañón para mantener al objetivo en la mira</li> </ul>
<b>Casos Alternativos</b>	<ul style="list-style-type: none"> <li>• El operador no puede mantener al objetivo en la mira</li> </ul>

*Tabla 8. Escenario de Leite – Derribar objetivo*

<b>Derribar objetivo</b>	
<b>Objetivo</b>	Derribar al objetivo antes de que realice su ataque
<b>Contexto</b>	El operador está apuntando al objetivo a través de la mira del cañón
<b>Recursos</b>	Cañón
<b>Actores</b>	Operador del cañón, Avión
<b>Set de Episodios</b>	<ul style="list-style-type: none"> <li>• El operador acciona el disparo del cañón</li> <li>• El avión es impactado por la munición del cañón</li> </ul>
<b>Casos Alternativos</b>	<ul style="list-style-type: none"> <li>• El operador no apunta adecuadamente y el blanco no es derribado</li> </ul>

*Tabla 9. Escenario de Leite – Respetar sector de responsabilidad*

<b>Respetar sector de responsabilidad</b>	
<b>Objetivo</b>	Mantener la mira del cañón dentro del sector de responsabilidad
<b>Contexto</b>	Ocurre durante toda la simulación
<b>Recursos</b>	Cañón
<b>Actores</b>	Operador del cañón
<b>Set de Episodios</b>	<ul style="list-style-type: none"> <li>• El operador mueve el cañón sin que la mira quede fuera del sector de responsabilidad</li> </ul>
<b>Casos Alternativos</b>	<ul style="list-style-type: none"> <li>• La mira del cañón queda fuera del sector de responsabilidad</li> </ul>

*Tabla 10. Escenario de Leite – Evaluar desempeño*

<b>Evaluar desempeño</b>	
<b>Objetivo</b>	Evaluar el desempeño del operador en una simulación
<b>Contexto</b>	El operador ha finalizado una simulación
<b>Recursos</b>	Gráficos
<b>Actores</b>	Instructor, Operador del cañón
<b>Set de Episodios</b>	<ul style="list-style-type: none"> <li>• El instructor visualiza gráficos con los resultados de la simulación</li> <li>• El instructor evalúa el desempeño del operador en la simulación</li> </ul>
<b>Casos Alternativos</b>	–

*Tabla 11. Escenario de Leite – Evaluar progreso*

<b>Evaluar progreso</b>	
<b>Objetivo</b>	Evaluar el progreso del operador a lo largo de las sesiones de simulación
<b>Contexto</b>	El operador ha realizado varias simulaciones a lo largo del tiempo
<b>Recursos</b>	Reporte
<b>Actores</b>	Instructor, Operador del cañón
<b>Set de Episodios</b>	<ul style="list-style-type: none"> <li>• El instructor visualiza un reporte con los resultados de las simulaciones a lo largo del tiempo.</li> <li>• El instructor evalúa el progreso del operador a lo largo de las sesiones de simulación</li> </ul>
<b>Casos Alternativos</b>	<ul style="list-style-type: none"> <li>• El instructor selecciona una de las simulaciones para visualizar el desempeño del operador en la misma</li> </ul>

*Tabla 12. Escenario de Leite – Administrar usuarios*

<b>Administrar usuarios</b>	
<b>Objetivo</b>	Administrar los usuarios del sistema
<b>Contexto</b>	El administrador desde efectuar cambios en los usuarios del sistema
<b>Recursos</b>	Gestor de usuarios
<b>Actores</b>	Administrador, Usuario
<b>Set de Episodios</b>	<ul style="list-style-type: none"> <li>• El administrador ingresa al gestor de usuarios</li> <li>• El administrador selecciona la acción que desea realizar</li> <li>• El administrador efectúa los cambios necesarios</li> <li>• El administrador confirma los cambios realizados</li> </ul>
<b>Casos Alternativos</b>	<ul style="list-style-type: none"> <li>• Si la acción es agregar, se añade un nuevo usuario al sistema</li> <li>• Si la acción es editar, los datos del usuario son modificados</li> <li>• Si la acción es desactivar un usuario activo, el usuario ya no puede ser utilizado en el sistema</li> <li>• Si la acción es activar un usuario desactivado, el usuario puede volver a ser utilizado en el sistema</li> <li>• Si la acción es eliminar, el usuario elegido es removido permanentemente del sistema</li> </ul>