

CAPÍTULO 2. Sistema de Simulación Dinámica de Movimiento para el Aprendizaje del Método Runge-Kutta.

<https://doi.org/10.64325/6bnkgg94>

Autores

Yuliana Andrea Trejos Bobadilla

Universidad de la Amazonia

<https://orcid.org/0009-0005-8559-3507>

Diva Natalia Vargas Tovar

Universidad de la Amazonia

<https://orcid.org/0009-0001-3595-7625>

Edwin Eduardo Millán Rojas

Universidad de la Amazonia

<https://orcid.org/0000-0002-4258-4601>

Como citar: Trejos Bobadilla, Y. A., Vargas Tovar, D. N., & Millán Rojas, E. E. (2025). Sistema de Simulación Dinámica de Movimiento para el Aprendizaje del Método Runge-Kutta. En E. E. Millán Rojas (Ed.), *Uso de la realidad Virtual 3D para articular la enseñanza de la Física en Ingeniería de Sistemas* (pp. 47-74). Editorial Didaxis. <https://doi.org/10.64325/6bnkgg94>

Resumen

La implementación y comprensión de métodos numéricos iterativos, el desarrollo de un sistema de simulación dinámica de movimiento basado en el método Runge Kutta presenta desafíos pedagógicos significativos en la enseñanza de métodos numéricos iterativos aplicados a un sistema físico complejo, se utilizó la metodología ágil eXtreme Programming (XP), para priorizar la eficiencia en el desarrollo del sistema, la interacción continua entre los desarrolladores y usuarios finales, permitió crear un sistema interactivo y preciso que además opera en tiempo real. El sistema incluye un motor de simulación de alto rendimiento que optimiza la velocidad en la que opera el sistema, la precisión numérica y la eficiencia algorítmica lo que permite parametrizar las trayectorias dinámicas, la arquitectura modular que presenta el proyecto permite la escalabilidad y flexibilidad, mientras que la interfaz facilita la manipulación dinámica de parámetros, el análisis de errores numéricos y la interpretación de resultados. El sistema permitirá a los usuarios realizar simulaciones con múltiples objetos, observar trayectorias en tiempo real y analizar datos detallados como pasos de integración, intervalos de tiempo y errores estándar, esto fomenta la comprensión de conceptos complejos y apoya

decisiones fundamentadas en contextos académicos. La implementación del método Runge Kutta en la simulación asegura la precisión en el cálculo de la trayectoria definida por el objeto y la adaptabilidad de parámetros, enriqueciendo la experiencia de aprendizaje en los usuarios, las futuras mejoras incluyen la incorporación de algoritmos para el análisis de datos, exportación de resultados y la visualización gráfica intuitiva. Este proyecto destaca la efectividad de los entornos virtuales como herramientas educativas, promoviendo la comprensión de los métodos numéricos a través de la interacción dinámica con el sistema.

Palabras Claves: Simulación dinámica; método Runge-Kutta; eXtreme Programming; métodos numéricos

Abstract

The implementation and understanding of iterative numerical methods, along with the development of a dynamic motion simulation system based on the Runge-Kutta method, present significant pedagogical challenges when teaching iterative numerical methods applied to complex physical systems. The agile methodology eXtreme Programming (XP) was employed, prioritizing development efficiency and continuous interaction between developers and end users. This approach enabled the creation of an interactive, accurate, and real-time operating system. The system features a high-performance simulation engine that optimizes execution speed, numerical precision, and algorithmic efficiency, allowing the parametrization of dynamic trajectories. Its modular architecture provides scalability and flexibility, while the user interface facilitates dynamic parameter manipulation, numerical error analysis, and result interpretation. The system enables users to perform simulations with multiple objects, observe trajectories in real time, and analyze detailed data such as integration steps, time intervals, and standard errors. This fosters a deeper understanding of complex concepts and supports informed decision-making in academic contexts. The implementation of the Runge-Kutta method in the simulation ensures precision in calculating object-defined trajectories and adaptability of parameters, thereby enriching the learning experience for users. Future improvements include the incorporation of data analysis algorithms, result export capabilities, and intuitive graphical visualization. This project highlights the effectiveness of virtual environments as educational tools, promoting the understanding of numerical methods through dynamic interaction with the system.

Keywords: Dynamic simulation; Runge-Kutta method; eXtreme Programming; numerical methods

Introducción

El proyecto de investigación tuvo como objetivo principal desarrollar un sistema de simulación dinámica de movimiento para el aprendizaje del método Runge-Kutta, enfocado en un entorno virtual interactivo, la implementación de herramientas virtuales facilita los procesos de enseñanza y aprendizaje beneficiando a estudiantes y docentes, además fomenta una interacción activa que permite a los usuarios desarrollar múltiples competencias de aprendizaje (Sabaduche, 2015).

El sistema se desarrolló dentro el contexto del programa de ingeniería de sistemas, teniendo énfasis en la enseñanza del método Runge-Kutta, contenido temático que pertenece a la asignatura de Métodos Numéricos de la universidad de la amazonia, la implementación del método en el sistema busca fortalecer el aprendizaje practico y conceptual de los estudiantes, mediante el uso de herramientas virtuales avanzadas adaptadas al entorno educativo.

Durante los últimos años, la educación ha evolucionado de manera significativa debido a los avances tecnológicos, el uso de herramientas virtuales ha transformado el proceso de enseñanza- aprendizaje tradicional, y ha fomentado la interactividad en la enseñanza haciendo que los estudiantes sientan más atracción por este método de enseñanza. Los estudiantes enfrentan diferentes desafíos en la adaptación y aprendizaje en la educación tradicional, desafíos que se buscan superar de tal manera que garantice una educación de calidad para los estudiantes (Robles *et al.*, 2022).

Este proyecto surge como respuesta a las dificultades que enfrentan los estudiantes y docentes en el proceso de enseñanza-aprendizaje de conceptos matemáticos y físicos en la educación tradicional, la falta de herramientas interactivas y accesibles, dificultan el aprendizaje de métodos con un alto nivel de complejidad como lo es el método de Runge-Kutta, la simulación de trayectorias y el análisis de variación de resultados al modificar parámetros como el número de pasos o intervalos, son parámetros que a simple vista no se pueden ejecutar en el método tradicional que limita la capacidad de aprendizaje y aplicación del método Runge-Kutta. Por lo tanto, el sistema busca abordar estas limitaciones, realizando una plataforma que facilite la

comprensión y el análisis de estos conceptos para mejorar la calidad de enseñanza en la Universidad de la Amazonia.

La metodología implementada en el proyecto de investigación fue la metodología eXtreme Programming (XP), metodología ágil que se enfoca en el trabajo colaborativo entre los actores involucrados en el sistema y la capacidad de adaptación a cambios durante el proceso de desarrollo del proyecto (Wang, Conboy & Cawley, 2012). Durante el proceso de elaboración del sistema, los desarrolladores y el cliente mantienen una comunicación constante para poder realizar mejoras al producto final y que este cumpla con las necesidades del usuario y el objetivo principal, crear un sistema de simulación dinámica de movimiento para el aprendizaje del método Runge-Kutta.

La herramienta virtual implementada para el diseño del proyecto fue FireStorm, un visor de código abierto que permite la creación de mundos virtuales, además, se utilizó visual paradigma para diseñar los diagramas implementados, que incluyen el diagrama de casos de uso, el diagrama de clases y el diagrama de despliegue, los cuales funcionan como guía para los desarrolladores de cómo está estructurado el sistema. Los diagramas UML (Lenguaje Unificado de Modelado) son representaciones graficas que permiten modelar un software, además facilita la comprensión y el diseño de los sistemas (Cortez, Páez & Simanca, 2021). Debido a la metodología que se aplicó a el proyecto, se realizaron diferentes ajustes obtenidos mediante la retroalimentación de los diagramas UML, lo cual garantizo las expectativas de los usuarios.

Como resultado se implementaron cuatro escenarios al sistema, cada uno con el objetivo de abordar el concepto del método Runge-Kutta, estos escenarios permiten que los usuarios comprendan mejor el tema, fomentando la participación de los estudiantes. Además, se aplicaron encuestas a los docentes que imparten la materia métodos numéricos en la universidad de la amazonia, donde se expresó una gran satisfacción por el desarrollo y uso de las simulaciones dentro del entorno virtual, también se ha demostrado que la incorporación de tecnologías interactivas en el aprendizaje contribuye eficazmente a la retención y comprensión de conceptos abstractos. El proyecto está organizado en secciones que describen la metodología empleada, los resultados obtenidos mediante la evaluación del sistema y las

conclusiones del proyecto, destacando la importancia de implementar el sistema de simulación dinámica de movimiento para el aprendizaje del método Runge-Kutta.

Materiales y métodos

En el desarrollo de software, la correcta elección de una metodología para la implementación de un sistema es importante para garantizar el éxito de un proyecto, especialmente cuando se trata de un entorno dinámico y cambiante, la implementación de la metodología Extreme Programming (XP) es una solución para proyectos que requieran de cambios y colaboración continua en el desarrollo, como en el caso de un Sistema de Simulación Dinámica de Movimiento utilizando el Método de Runge-Kutta donde se fundamenta en los principios de la metodología Extreme Programming (XP), que se centra en la adaptabilidad y la mejora continua.

Según la investigación realizada por Beck & Andrés (2004), la metodología XP facilita a los desarrolladores del software realizar cambios a los requisitos del sistema en cualquier fase de proyecto, lo que es fundamental en un proyecto donde las variables pueden cambiar debido a nuevas condiciones, además, Extreme Programming permite una constante comunicación y colaboración entre los actores implicados en el proyecto, lo que es fundamental para asegurar el éxito del sistema. La comunicación continua entre los actores les facilita a los desarrolladores realizar ajustes en el sistema de manera oportuna, cumpliendo con el objetivo principal y la satisfacción de los usuarios (Beck & Andrés, 2004).

La metodología XP promueve la responsabilidad y la transparencia durante la fase de construcción del proyecto, por lo que los miembros del equipo de desarrollo se sienten comprometidos con el correcto funcionamiento del proyecto, lo que garantiza que el sistema termine sea eficiente, y se adapte a los cambios constantes que requiera el usuario final. La metodología XP enfatiza en la importancia de realizar pruebas continuas al sistema, lo que permite adaptar el sistema a constantes cambios, asegurar la calidad y el correcto funcionamiento del software (Letelier & Penadés, 2006). Para la construcción del proyecto se implementó el visor 3D “Firestorm”, un visor de código abierto que permite el acceso al servidor “UdlaVerso”, herramienta en la cual se

desarrolló el proyecto, además para la fase de diseño, se implementó la herramienta Visual Paradigma Online, para realizar los diagramas de clases, de despliegue y de casos de uso, importantes para la estructuración del proyecto.

Fase análisis

La siguiente tabla muestra los contenidos de la materia de métodos numéricos, donde se evidencia el método Runge-Kutta y otros contenidos afines. El contenido temático abordado en este proyecto de investigación es fundamental ya que permite aproximar la solución de estas ecuaciones que, en muchos casos, no pueden resolverse de manera exacta, mediante cálculos iterativos

Tabla 1. *Contenido temático métodos numéricos, unidad 4*

unidad 4: Integración y derivación numérica	descripción
Integración numérica	Descripción de la integración numérica y su aplicación con el método Runge-Kutta.
Derivación numérica	Descripción de la derivación numérica evaluando la función en puntos cercanos.
Ecuaciones diferenciales ordinarias	Descripción de las ecuaciones diferenciales ordinarias y su resolución mediante el método Runge-Kutta.

Nota. Contenido temático del área de métodos numéricos, unidad 4. Fuente: Universidad de la Amazonia.

A partir de esta temática, se desarrolló una investigación detallada con el fin de profundizar en el tema y así poder realizar un proyecto más participativo y claro, una vez realizada la investigación sobre el tema, se utilizó como método de recolección de datos la entrevista, la cual permitió obtener información detallada de las personas involucradas en el proyecto de investigación (Diaz *et al*, 2013). La entrevista realizada a el docente, se implementaron diferentes preguntas que permitieron conocer acerca del tema que se quería abordar en el proyecto, al igual que con la entrevista realizada a el tutor de los desarrolladores, a continuación, se visualizan las preguntas formuladas y la estructura diseñada para evaluar y aplicar conceptos teóricos.

Tabla 2. *Entrevista aplicada a el docente.*

1.	¿Tiene usted conocimiento acerca de la herramienta de Udla Verso (en caso de que no sepa, explicarle de que se trata)
2.	¿Cómo enseñas actualmente conceptos relacionados con el movimiento de objetos o sistemas dinámicos?
3.	¿Qué opinas sobre el uso de herramientas computacionales para enseñar y analizar trayectorias en tiempo real?
4.	¿Qué recursos digitales o simuladores usas en tus clases para enseñar conceptos matemáticos o científicos?
5.	¿Cómo integrarías un sistema de simulación como este para que los estudiantes comprendan mejor el movimiento de objetos y su análisis?

Nota. Entrevista desarrollada y aplicada docente que maneja la asignatura de métodos numéricos. Fuente: elaboración propia.

Una vez formuladas las preguntas para la entrevista, estas fueron presentadas a un experto en el desarrollo de campus virtual 3D, específicamente en la plataforma "UdlaVerso" (código interno No. 600.6.716, Acta 010 del 02 de noviembre de 2023), cuyo objetivo es articular el proceso misional de docencia en la Universidad de la Amazonia. El propósito de esta validación fue comprobar si las preguntas que se formularon cubren adecuadamente las necesidades del usuario.

La entrevista fue realizada a través de WhatsApp con el docente encargado de dictar la materia de métodos numéricos. Dado que el docente no estaba familiarizado con el proyecto "UdlaVerso" se le explicó previamente el concepto de este, y luego se procedió a realizar la entrevista. Posteriormente, los datos obtenidos fueron organizados en lenguaje natural, con el fin de facilitar su análisis y describir de manera clara cada aspecto relevante para el desarrollo del módulo interactivo. Este proceso aseguró que el diseño del prototipo de realidad aumentada se adapte a las necesidades tanto de los estudiantes como de los docentes expertos en la materia, garantizando su funcionalidad y atractivo.

Fase de construcción

La fase de construcción comenzó tras completar la fase de análisis y el levantamiento de requisitos, esta etapa, se enfocó en el desarrollo y diseño de los componentes que integran el proyecto de investigación, implementando las funcionalidades principales del sistema, para posteriormente llevar a cabo pruebas y garantizar que el sistema opera de manera eficiente, la integración de herramientas tecnológicas de aprendizaje mejoró la accesibilidad y la interacción en el proceso educativo, superando las limitaciones físicas de la enseñanza tradicional, también, las herramientas tecnológicas fomentaron la participación activa de los estudiantes, promoviendo el aprendizaje colaborativo (Tzenguzha *et al*, 2021). También, cada fase de construcción en el sistema de simulación dinámico fue documentado, y se implementaron cambios pertinentes para la satisfacción de los usuarios.

Durante la fase de construcción, se implementó el diagrama de casos de uso, el cual es un lenguaje de modelado unificado que permite la visualización de las relaciones entre los actores involucrados para conocer las acciones a realizar por cada uno de estos actores dentro del proyecto, para realizar el diagrama de casos de uso se parte de los requisitos funcionales que fueron creados en la fase de análisis del sistema y enfocándose desde el punto de vista de los usuarios (Arias, 2016). Una vez diseñado el diagrama, se realizan las especificaciones de casos de uso, que describen detalladamente la interacción de los usuarios con el sistema e incluyen información sobre los pasos a seguir para cada uno de los casos de uso, permitiendo detectar inconsistencias o problemas antes de implementarse en el sistema. Adicionalmente, se detallaron los actores involucrados, los requisitos asociados y los resultados esperados en cada interacción con el sistema, asegurando un diseño claro y funcional.

Posteriormente, se implementó el diagrama de clases, uno de los diagramas estructurales UML que permite visualizar la estructura del sistema con sus clases, atributos, métodos, cardinalidades y las relaciones que existen entre las clases, este tipo de diagrama representa los recursos necesarios para la construcción y operabilidad del sistema (Vidal *et al*, 2014). Su construcción es clave para identificar como se asocia, se hereda o se compone un sistema, esto facilita la comprensión y comunicación con los

demás actores involucrados, dentro del proyecto de investigación, en el marco del proyecto de investigación el diagrama de clases es fundamental para identificar los objetos que requieren interactividad.

Además, se diseñó el diagrama de despliegue, un diagrama estructurado que representa la arquitectura del sistema desde la perspectiva de la distribución de los artefactos de software, permitiendo representar los elementos en el físicos y lógicos (Sayas y Garcia, 2014). Este diagrama de despliegue representa de manera clara como se distribuyen los componentes del sistema en la infraestructura de hardware, como los servidores y los equipos, también se especificó las tecnologías de comunicación que se implementaron, como aplicaciones, base de datos o configuraciones, que permitieron garantizar que el entorno de despliegue sea capaz de soportar el sistema de manera eficiente.

Adicionalmente, se realizó un boceto a mano alzada basado en las entrevistas realizadas a los docentes, lo que permitió organizar las ideas y visualizar los diferentes escenarios que se implementaron en el proyecto. Este proceso fue fundamental, ya que, a partir de los bocetos, se logró obtener una representación más precisa de cómo se visualizaría la simulación dinámica de movimiento para el aprendizaje del método Runge-Kutta en el entorno de realidad aumentada. El proceso de construcción de los escenarios inicio con la elaboración de la estructura denominada piscina, que representa el entorno general donde se ejecutara el proyecto, posteriormente se incorporaron los equipos donde el usuario seleccionara el objeto con el que quiere marcar la trayectoria, de esta manera se logra obtener una representación precisa en términos de organización de los escenarios.

Fase evaluación

En esta fase de evaluación, se empleó el visor Firestorm, para acceder al 'UdlaVerso' una herramienta de código abierto, la cual requirió de un proceso de instalación, el visor se descargó desde el sitio oficial de Firestorm y se completó la configuración siguiendo el asistente de instalación, habilitando el inicio de sesión, para

así acceder al entorno, por medio de las credenciales que fueron asignadas con anticipación por el ingeniero a cargo.

Además, se realiza un análisis detallado de los resultados obtenidos a lo largo del desarrollo, con el objetivo de evaluar la calidad del producto desarrollado y su alineación con las expectativas de los usuarios. Esto incluyó una revisión exhaustiva de la plataforma interactiva, observando si los usuarios eran capaces de comprender y aplicar los conceptos físicos y matemáticos a través de la simulación. Se verificó que la herramienta sea intuitiva, fácil de usar y, lo más importante que realmente mejorara el proceso de aprendizaje en los estudiantes, a partir de los resultados obtenidos durante la evaluación se pudieron realizar ajustes y mejoras necesarias para optimizar el funcionamiento del sistema y asegurar que se cumpla con el propósito educativo.

Resultados

En esta sección se presentan los resultados obtenidos a lo largo del desarrollo de proyecto “simulación dinámica de movimiento para el aprendizaje del método Runge-Kutta” el cual implementó la metodología XP en el contexto de la educación interactiva. Se incluye un análisis detallado de los datos recopilados mediante las encuestas aplicadas a los docentes encargados de la asignatura de métodos numéricos, así como un análisis exhaustivo de los diseños y la construcción de los escenarios. A continuación, se presentan los resultados de las fases anteriormente mencionadas.

Resultados de la fase de análisis

La temática seleccionada se presenta de manera detallada en la tabla 2.1, que describe la unidad 4 de la asignatura Métodos Numéricos en el programa de Ingeniería de Sistemas de la Universidad de la Amazonia. Esta unidad fue utilizada como base para la creación de la “simulación dinámica de movimiento para el aprendizaje del método Runge-Kutta” con el objetivo de facilitar el aprendizaje de los estudiantes. Tras realizar la investigación exhaustiva sobre el tema, se procedió a realizar una encuesta a el docente encargado de dictar la materia métodos numéricos, el formulario consolidado para la entrevista se encuentra en la tabla 2.

A continuación, se presentan las respuestas obtenidas través de la entrevista realizada al docente encargado, destacando los aportes claves que contribuyeron al desarrollo del proyecto.

Tabla 3. *Respuesta de la entrevista*

1. ¿Tienes conocimiento de que se trata el UdlVerso? (En caso de que no esté familiarizado, me gustaría explicarte brevemente de qué se trata). Un estudiante de otro semestre lo socializó, al igual que participe del congreso que se realizó en el Udlaverso.
2. ¿Cómo enseñas actualmente conceptos relacionados con el movimiento de objetos o sistemas dinámicos? Actualmente modelamos aplicaciones como trayectoria de planetas alrededor del sol, trayectoria de satélites, tractores biológicos (fermentadores), trayectorias de balones de fútbol, clavado entre otras muchas aplicaciones.
3. ¿Qué opinas sobre el uso de herramientas computacionales para enseñar y analizar trayectorias en tiempo real? Deben usarse los recursos disponibles incluida la inteligencia artificial, software de análisis numérico como Python, ambientes de trabajo colaborativo como Google Colab, ambientes de realidad aumentada para disfrutar más de la experiencia de aprender.
4. ¿Qué recursos digitales o simuladores usas en tus clases para enseñar conceptos matemáticos o científicos? Uso Python, y Google Colab. Construimos los simuladores a partir de la solución de sistemas de ecuaciones diferenciales.
5. ¿Cómo integrarías un sistema de simulación como este para que los estudiantes comprendan mejor el movimiento de objetos y su análisis? ¿Cómo se integra? No tengo mucha idea de la implementación, pero como usuario o estudiante sería ideal que el ambiente de simulación permitiera generar cambios en las variables de entrada como velocidad inicial del objeto (U objetos) ángulo, masas. Y que mostrará la animación de la solución mediante un ambiente gráfico 3D de la trayectoria.

Nota. Entrevista y respuesta desarrollada y aplicada al docente que maneja la asignatura de métodos numéricos. Fuente: elaboración propia.

Tras aplicar la encuesta al docente, se realizó un análisis detallado para identificar las necesidades específicas de los usuarios respecto el aprendizaje del tema propuesto, el objetivo de este análisis fue recolectar los requisitos necesarios para satisfacer a los estudiantes, asegurando el correcto funcionamiento y la eficiencia del proyecto “simulación dinámica de movimiento para el aprendizaje del método Runge-Kutta”. A continuación, se detallan los requerimientos funcionales y no funcionales identificados a partir del análisis de las respuestas obtenidas durante la entrevista con

el docente. Estos requerimientos representan elementos clave en el desarrollo de la plataforma educativa, ya que encapsulan las necesidades específicas y expectativas del usuario final, garantizando que el diseño y la implementación del sistema se alineen con los objetivos pedagógicos y operativos planteados. Los requisitos funcionales, son especificaciones que determinan el comportamiento y la funcionalidad que debe tener un sistema para cumplir con las necesidades de los usuarios (Pradel y Raya, 2010). A continuación, se presentan los requisitos funcionales.

Tabla 4. Requerimientos Funcionales

Id	Nombre	Descripción	Origen
RF001	Objetos de simulación	El administrador deberá crear múltiples objetos que el usuario podrá seleccionar y realizar la simulación.	E1_001
RF002	Seleccionar Rampa	El usuario tendrá una opción en el sistema para escoger la altura de la rampa que desea manejar.	E1_001
RF003	Objeto impulsor	El sistema contará con un objeto en cada rampa que tendrá como objetivo impulsar el objeto que el usuario seleccione para realizar la simulación.	E1_001
RF004	Numero de pasos representaci ^o n del movimiento	El sistema debe permitir que el usuario seleccione por medio de un monitor el número de pasos con el que se representara la simulación.	E1_001
RF005	Simulación del movimiento Runge-Kutta	El sistema simulara el movimiento del objeto seleccionado utilizando el método de Runge-Kutta para describir su trayectoria.	E1_001
RF006	Visualización de datos en tiempo real	El sistema debe proporcionar al usuario datos en tiempo real, como el paso, el número de intervalos, la altura, el número de operaciones, el número total de puntos y el error estándar dentro de la simulación.	E1_001
RF007	Simulación de objetos	El sistema permitirá al usuario realizar simulaciones con varios objetos para observar como el número de pasos afecta a las demás variables dentro de la simulación.	E1_001

Nota. Requisitos funcionales. Fuente: elaboración propia.

Luego de extraer los requisitos funcionales que se detallan en la tabla 4 para el desarrollo del proyecto, se procede a elaborar las especificaciones de requerimientos en el formato de IEEE para realizar un análisis detallado a cada uno de los requisitos, definiendo de manera precisa las funcionalidades y comportamientos que debe tener el sistema, de esta manera se asegura que el proyecto cumpla con las expectativas y

necesidades del usuario. A continuación, se presenta un ejemplo de especificación de requerimiento, lo cual permitirá una mejor visualización y comprensión de estos para el desarrollo el sistema

Tabla 5. *Especificación de requerimientos*

Especificación de requerimiento	
Código	R-001
Nombre	Objetos de simulación
Descripción	El administrador tendrá creado una variedad de objetos que permita hacer la simulación
Prioridad	Alta
Origen	Requisito del administrador
Usuarios	Administrador

Nota. Especificación de requerimientos. Fuente: modificado de IEEE.

A continuación, se presentan los requisitos no funcionales, según (Pérez *et al*, 2016), los requisitos no funcionales son las características técnicas que debe tener un software, para que este funcione de manera segura.

Tabla 6. *Requerimientos No Funcionales*

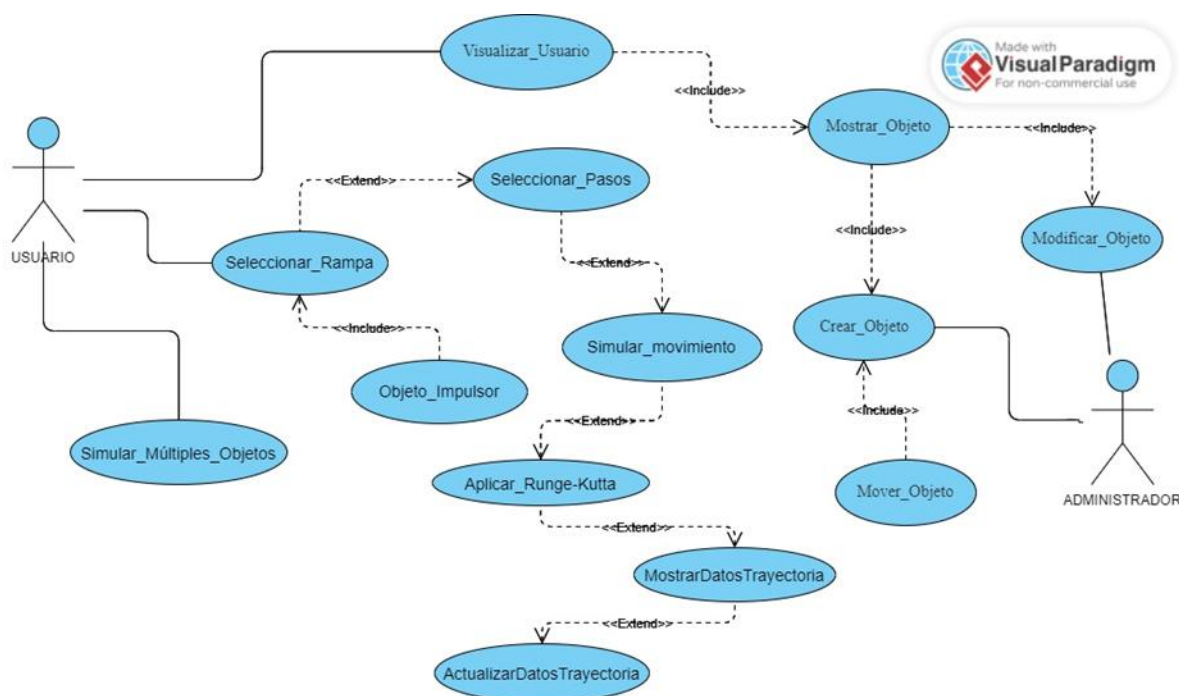
ID	NOMBRE	DESCRIPCIÓN
RF001	Rendimiento	El sistema debe ser eficiente, debe generar respuestas rápidas y eficientes para el usuario sin retrasos.
RF002	Usabilidad	El sistema debe ser eficaz e interactivo, con guías para que el usuario pueda manejar el sistema.
RF003	Seguridad	El sistema contará con seguridad para proteger los datos de los usuarios y las simulaciones que genere dentro del sistema.

Nota. Requerimientos No Funcionales. Fuente: modificado de IEEE.

En la fase de construcción del proyecto, se realiza la definición de los componentes del sistema y el diseño para este mismo, donde los usuarios podrán tener conocimiento acerca de cómo funciona el sistema, teniendo en cuenta los datos que se obtuvieron durante la fase de análisis del proyecto, la fase de construcción es importante para establecer e identificar como se integran los componentes del sistema, para garantizar que se cumpla con el objetivo principal del proyecto, luego se incorporan los diagramas UML a la fase de construcción para visualizar la estructura del software, lo que permitirá identificar las funcionalidades y los alcances del sistema. El primer diagrama que se implementará al proyecto es el diagrama de casos de uso, diagrama UML que es una representación gráfica del sistema donde los actores involucrados en el proyecto interactúan y se relacionan, los diagramas de caso de uso se modelan

basándose en los requerimientos funcionales obtenidos durante la fase de análisis y que se implementan desde la perspectiva del usuario (Arias, 2006). A continuación, se implementó el desarrollo del diagrama de casos de uso elaborado en la herramienta Visual Paradigm Online.

Figura 1. Diagrama de casos de uso



Nota. Diagrama de casis de uso. Fuente: elaboración propia en Visual Paradigm.

El diagrama de caso de usos se representa en la figura 1, donde los casos de uso se representan mediante óvalos que identifican las funcionalidades del sistema, para la elaboración del proyecto se identifican dos actores principales que son el usuario y el administrador los cuales tienen roles distintos dentro del sistema, el usuario puede interactuar con el funcionamiento del sistema, es decir puede realizar las simulaciones del movimiento del objeto, visualizar la trayectoria mientras que el administrador podrá crear y modificar objetos, lo que permite visualizar que el sistema funciona diferente dependiendo del rol con el que se ingrese al sistema cumpliendo con las expectativas del usuario.

El diagrama representa las relaciones entre los casos de uso, implementando las relaciones dependiendo de si se relaciona por inclusión, el cual indica que un caso de uso depende completamente de otro para completarse, o por extensión que indica que las relaciones son funcionalidades opcionales o adicionales. Este diagrama de casos de uso se utilizó en el desarrollo del proyecto para definir y visualizar los requisitos del

sistema desde la perspectiva del usuario, lo que permitió identificar y organizar las funcionalidades del sistema, las cuales se detallan a continuación.

Una vez diseñado el diagrama de casos de uso, se procedió a realizar un análisis detallado por medio de la especificación de casos de uso para cada uno de ellos, este análisis permitió definir con claridad las acciones y las funcionalidades que los actores podrán realizar dentro del sistema. La elaboración de las especificaciones permitió identificar las necesidades de los usuarios y la solución que el sistema ofrece, también proporciona un marco estructurado para verificar si el sistema cumple con los objetivos que se establecieron.

Tabla 7. Especificación de casos de uso

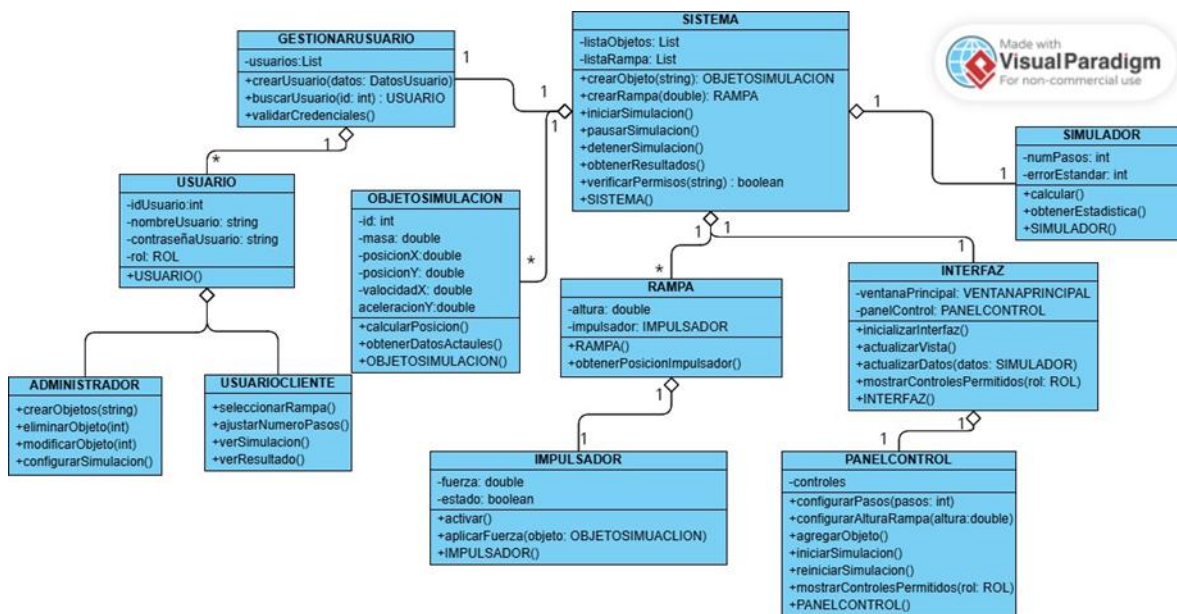
CU-<01>	CrearObjeto	
Versión	1.0	
Autores	Diva Vargas Tovar Yuliana Andrea Trejos Bobadilla	
Fuentes	R-001	
Actores:	Sistema	
Objetivos Asociados	Permitir la creación de objetos con un número variable de pasos. Casos de uso Asociados: Ninguno	
Casos de uso Asociados	Consultar_objeto	
Descripción	Permite crear un nuevo objeto con un número variable de pasos.	
Precondición	Ninguna	
Atributos	Número de objetos a crear	
Secuencia normal	Paso	Acción
	1	El sistema solicita el número de objetos a crear.
	2	El sistema solicita el número de pasos para cada objeto.
	3	El sistema crea los objetos con los números de pasos especificados.
	4	El sistema almacena los objetos creados.
Postcondición	Los objetos han sido creados y almacenados en el sistema	
Excepciones	Paso	Acción
		Ninguna
Rendimiento	Paso	Tiempo máximo
Rendimiento Importancia		
	Alta	
Urgencia	Alta	
Estado	Propuesto	
Estabilidad	Estable	
Comentarios	N/A	

Nota. Formato especificación de casos de uso, Fuente: modificado de IEEE.

Resultados de la fase de construcción

A continuación, se diseñó el diagrama de clases, el cual organiza los componentes del sistema y facilita la comprensión de las interacciones entre ellos, este diagrama se desarrolló a partir de los requerimientos establecidos en la fase de análisis y las especificaciones de caso de uso, permitiendo identificar los objetos necesarios para garantizar un desarrollo estructurado. Este diagrama describe los elementos principales del sistema, como las clases, los atributos, los métodos y las relaciones. A continuación, se presenta el diagrama de clases desarrollado para el proyecto.

Figura 2. Diagrama de clases



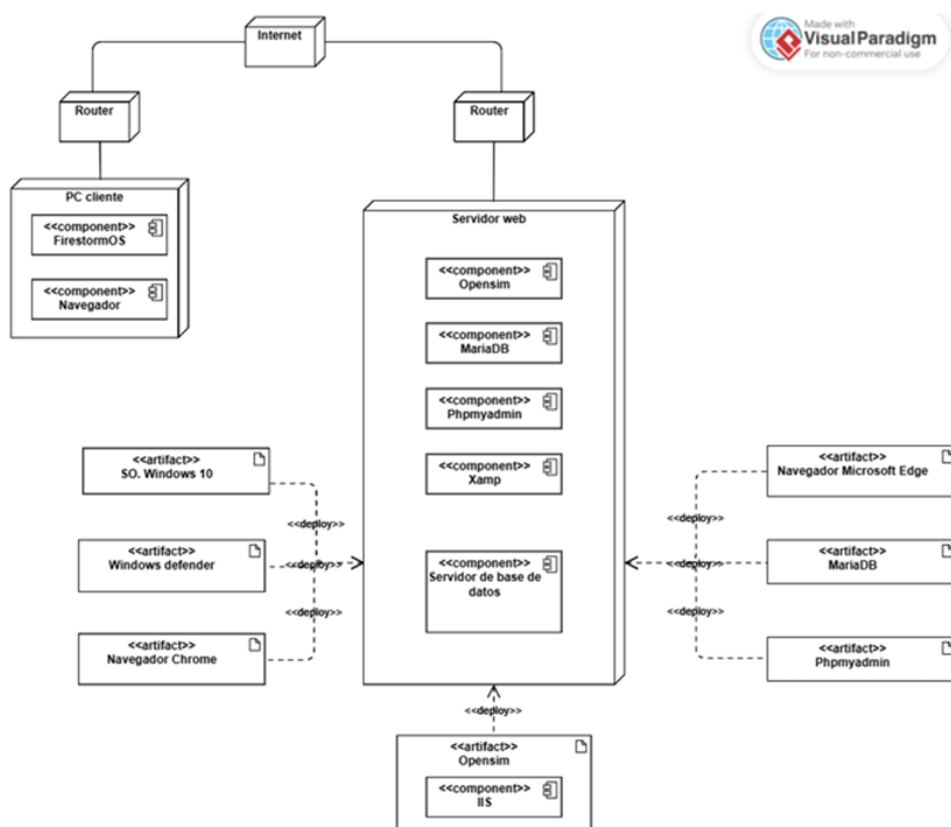
Nota. Diagrama de clases. Fuente: elaboración propia con la herramienta Visual Paradigm.

En el diagrama de clases, las clases están representadas por rectángulos, los cuales permiten crear objetos, agrupando los atributos que definen las propiedades del objeto y los métodos que indican las acciones que se pueden realizar, las relaciones entre las clases se representan mediante líneas que muestran cómo interactúan las clases entre sí, existen diferentes tipos de relaciones, entre ellas las relaciones de asociación y de herencia, las relaciones por asociación conectan clases de manera

genérica, mientras que la herencia, que es representada por un triángulo, indica que una clase hereda atributos y métodos de otra clase.

Además, se especifica la cardinalidad, que indica cuantos objetos de una clase están relacionados con los objetos de otra. El diagrama de clases se utiliza para modelar la estructura del sistema, lo que ayuda a tener una visualización de los componentes del sistema y como se relacionan entre sí, ya que proporcionan una forma clara y organizada de representar la arquitectura del sistema. Asimismo, se desarrolló el diagrama de despliegue, el cual permite visualizar la distribución de los componentes de software y hardware dentro del proyecto. Este diagrama es fundamental para comprender la estructura física y lógica del sistema, por medio de este diseño, se pueden conocer los nodos involucrados, los servidores, las conexiones y los protocolos empleados para garantizar una buena comunicación.

Figura 3. Diagrama de despliegue.



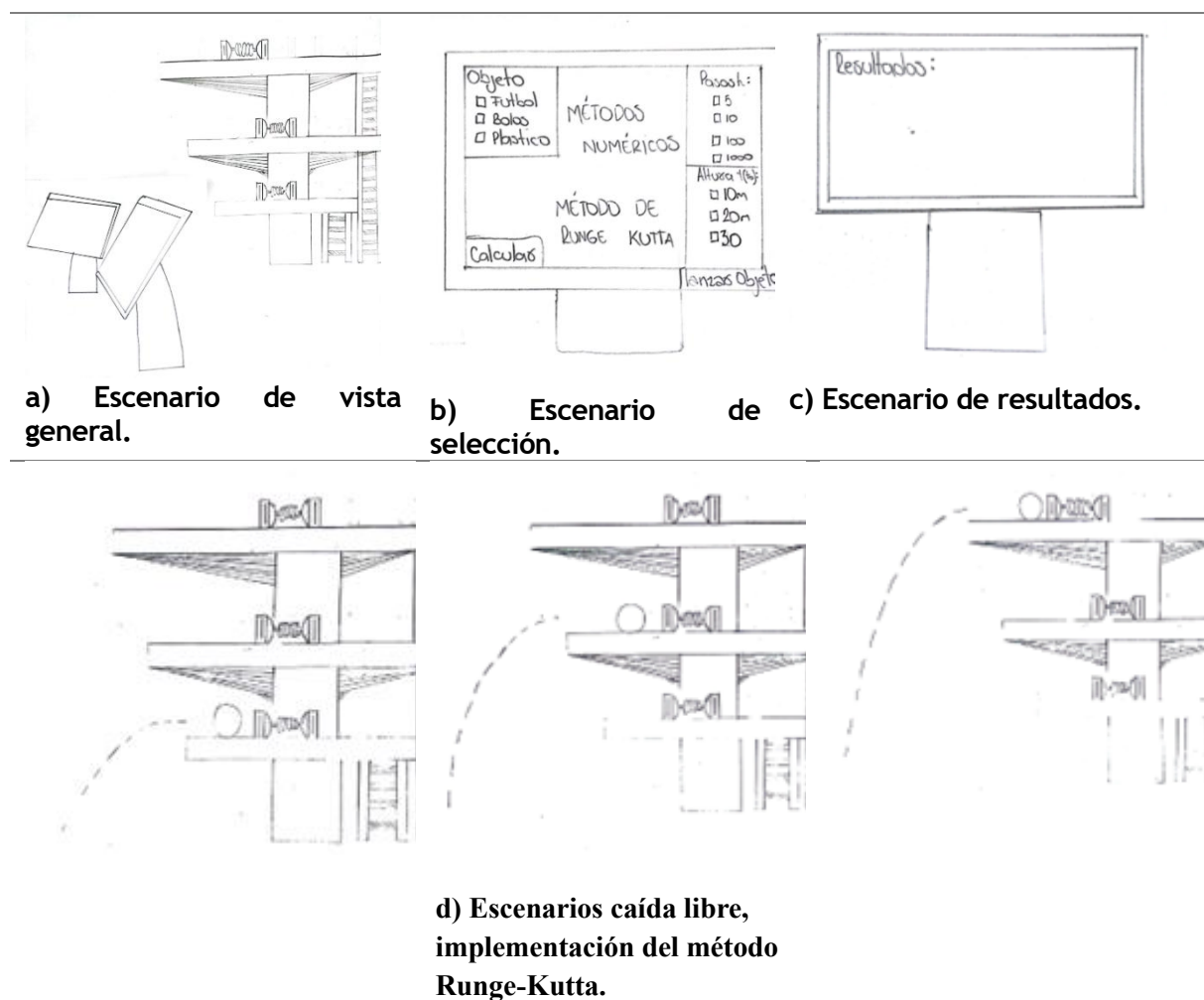
Nota. Diagrama de despliegue. Fuente: desarrollo propio usando la herramienta Visual Paradigm online.

En el diagrama de despliegue, representado en la figura 3, se identifican los nodos principales: PC cliente, servidor web e internet, el PC cliente representa el dispositivo que utiliza el usuario para interactuar con el sistema, este nodo incluye componentes como el sistema operativo y navegador web, que facilitan la conexión al servidor. El servidor web gestiona las solicitudes enviadas por los usuarios y contiene los principales servicios necesarios para el funcionamiento del sistema, los componentes del servidor incluyen Opensim, el encargado de manejar las conexiones mediante protocolos SSL/TLS, MariaDB, la base de datos utilizada para almacenar la información del sistema, Phpmyadmin, es el encargado de proporcionar una interfaz gráfica para la administración de la base de datos MariaDB.

Además, el servidor utiliza una plataforma XAMPP para integrar servicios como Apache y PHP, necesarios para el desarrollo y ejecución de aplicaciones web, por último, el servidor web contiene un componente IIS, que sirve para gestionar las solicitudes web y las conexiones. El internet actúa como intermediario entre el cliente y el servidor, permitiendo la comunicación entre ambos nodos, por medio de un router, reflejando como el sistema está distribuido físicamente por una red y como se conectan en un entorno real los dispositivos, las relaciones entre los nodos y componentes se representan por medio de líneas y flechas, las cuales indican las conexiones directas o dependencias entre los elementos. El diagrama de despliegue se utiliza para visualizar como están organizados los componentes del sistema y como interactúan entre sí.

En esta fase de proyecto también se diseñaron bocetos, los cuales permiten a los usuarios obtener una visión más clara de cómo se visualizara gráficamente el sistema, estos bocetos se elaboraron a mano, basándose en los requisitos funcionales del proyecto y los diagramas UML implementados con anterioridad en el sistema, la creación de los bocetos permitió a los diseñadores realizar una disposición adecuada de los elementos con los que los usuarios interactúan dentro del sistema de simulación, permitiendo tener una organización coherente y funcional, facilitando una experiencia de aprendizaje de calidad. A continuación, se implementaron los bocetos de la plataforma interactiva.

Figura 4. Bocetos iniciales de los escenarios.



Nota. Bocetos iniciales del sistema. Fuente: elaboración propia.

A continuación, se elaboraron los escenarios diseñados en la Figura 5, en el entorno virtual, el diseño del sistema fue creado en un entorno acuático centrado en una piscina que refleja un ambiente sereno, este entorno incluye elementos visuales como palmeras y los trampolines desde donde se ejecutara el procedimiento de caída libre para el aprendizaje del método Runge-Kutta. En este primer escenario, el usuario podrá observar todas las plataformas montadas y las plataformas con las cuales el usuario podrá interactuar.

Figura 5. *Escenario de vista general.*



Nota. Elaboración de los objetos 3D iniciales de las zonas definidas. Fuente: elaboración propia.

El siguiente escenario permite a los usuarios interactuar con el monitor, seleccionando los datos necesarios para realizar la simulación, como el objeto, la altura y el número de pasos, con el fin de realizar los diferentes cálculos correspondientes de la simulación.

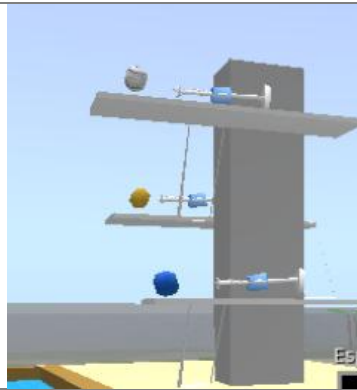
Figura 6. *Escenario de selección.*



Nota. Selección de variables de simulación. Fuente: elaboración propia.

El siguiente escenario, correspondiente a la caída libre, se implementará el método Runge Kutta. Una vez que el usuario haya seleccionado las variables en el escenario anterior, los objetos seleccionados se ubicarán en la plataforma teniendo en cuenta la altura y el número de pasos que se seleccionó, el usuario tendrá la opción de presionar un botón llamado “calcular” que iniciaría con la simulación.

Figura 7. *Escenario de simulación.*



Nota. Escenarios caída libre, implementación del método Runge-Kutta. Fuente: elaboración propia.

Por último, el escenario de resultados, una vez el usuario seleccione el botón de “calcular” en la pantalla de resultados aparecerán todos los datos que se obtuvieron al realizar la simulación de caída libre con las variables que selecciono el usuario.

Figura 8. *Escenario de resultados.*



Nota. Escenarios de resultados. Fuente: elaboración propia.

En el área de programación, para ejecutar adecuadamente la simulación e implementar la temática que se está trabajando, se procede después de la creación de todos los escenarios en “FireStorm” la implementación del script correspondiente a cada objeto, teniendo en cuenta que cada uno cumple con una función diferente.

Dando ejemplo a uno de los scripts más importantes partiendo de la creación de un monitor en 3D que tiene como objetivo facilitar la interacción del usuario con la simulación, ya que es el encargado de la elección de la altura, el objeto y el número de pasos, variables importantes para el funcionamiento de la simulación. En esta se debe

declarar los objetos “Balón Futbol, Balón Bolos, Balón Voleibol” dentro del entorno virtual para poder ser llamados y así mismo poder enlazarlos con “altura” y “números de pasos”, ya que no puede ejecutarse completamente hasta obtener todos los datos.

Ejemplo del algoritmo de ejecución del objeto

```
integer menuChanne1= -12345; // Canal del menú
key userKey; // Variable para almacenar la clave UUID del usuario
integer menuHandler;
Default
{
State_entry()
{
llistenRenove(menuHandler); // Elimina cualquier escucha previa
MenuHandler=lllisten(menuChanne1,"", NULL_KEY, "").
}
touch_star(integer total_number)
{
// Almacena la clave UUIO del usuario que tocó el objeto
userKey =llDetectedKey(0);
// Abre el cuándo se toca el objeto
llDialog(userKey, “selecciona la altura:” [“10 metros”, “20 metros”, “30 metros”],
menuChannel);
}
listen (integer channel, string name, key id, string message)
{
{
if (channel == menuChanel)
// Maneja las opciones del menú
if (message == "10 metros")
{
llOwnerSay(“has seleccionado 10 metros. Selecciona el número de pasos:”)
llDialog(userKey, “selecciona el número de pasos: [“5”, “10”, “15”], menuChannel);
llRegionSay(12345, “lanzar”);
else if (message==”20 metros”)
{
llOwnerSay(“has seleccionado 20 metros. Selecciona el número de pasos:”)
llDialog(userKey, “selecciona el número de pasos: [“5”, “10”, “15”], menuChannel);
llRegionSay(123456, “lanzar”);
}
}
```

```

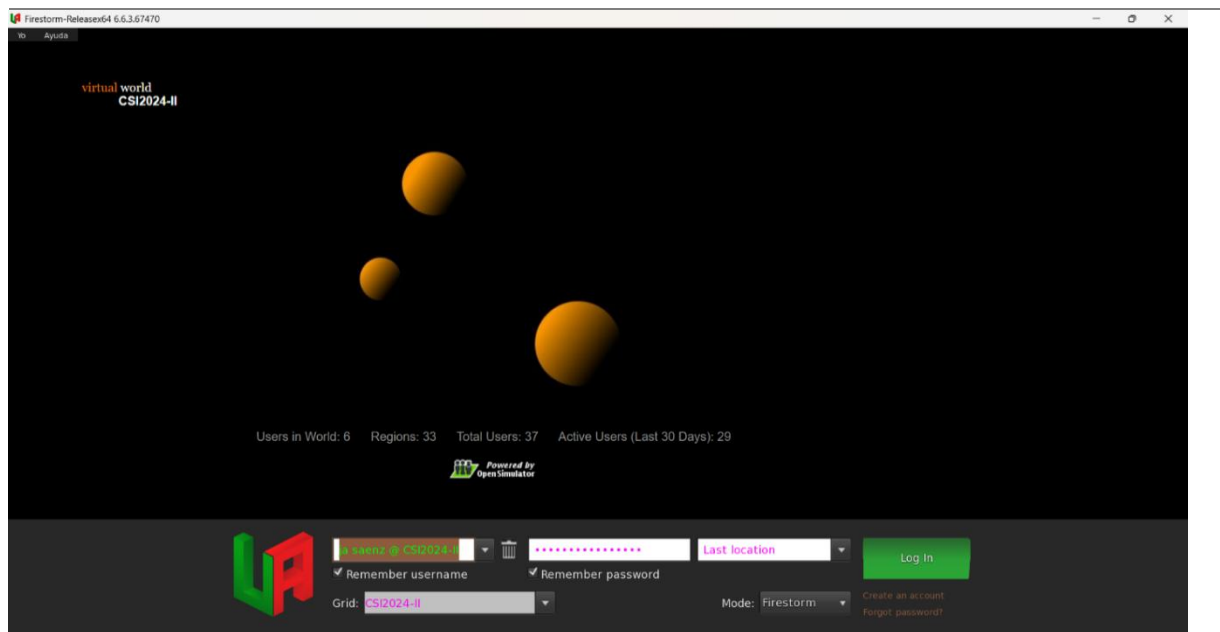
else if (message=="30 metros")
{
llOwnerSay("has seleccionado 30 metros. Selecciona el número de pasos:")
llDialog(userKey, "selecciona el número de pasos: ["5", "10", "15"], menuChannel);
llRegionSay(1234567, "lanzar");
}
{
else if (message=="5" || message=="10" || message=="15")
llOwnerSay("has seleccionado" + message+ "pasos");
}
}
}
}
}

```

Resultados de la fase de evaluación

El uso de plataformas virtuales según la investigación realizada por (Velasategui, 2017), se han establecido como una herramienta clave en las universidades, permitiendo que los docentes y estudiantes participen activamente en los procesos educativos, mediante el uso de tecnologías se facilita el acceso a la información, la comunicación y la creación de conocimientos, favoreciendo el aprendizaje de los estudiantes de manera integral. En la fase evaluativa, se realizó la instalación del visor FireStrom, una herramienta fundamental para el funcionamiento del sistema, también se configuraron de manera detallada los accesos, que se realizaron mediante la asignación de usuarios y contraseñas para cada participante, permitiendo que los usuarios puedan acceder a la plataforma de manera segura y eficiente. También se aseguró que cada participante tuviera los permisos necesarios para interactuar con la plataforma.

Figura 9. Resultado de la instalación del visor FireStorm.



Nota. Instalación y despliegue del visor FireStorm. Fuente: Interfaz principal del Visor Firestorm.

El análisis que se desarrolló al sistema demostró la importancia del uso de la herramienta Firestorm como visor para la ejecución de la simulación, demostró la eficiencia y la facilidad de uso de la plataforma con la integración del sistema que se desarrolló, los usuarios interactúan de manera dinámica con el sistema lo que permitió comprender las funciones y características de este mismo, las pruebas realizadas permitieron realizar mejoras en aspecto como la optimización de respuesta y en la interfaz para que los usuarios tuvieran una experiencia satisfactoria con el sistema.

La evaluación del sistema se desarrolló mediante actividades específicas, es decir se determinaron la implementación de algunas simulaciones para evaluar la estabilidad de la plataforma, el resultado de esta evaluación demostró que la plataforma cumplió con los requisitos y diseños establecidos en la fase de análisis y la fase de construcción del proyecto, lo que permitió garantizar una experiencia de aprendizaje eficiente y accesible para todos los usuarios.

Conclusiones

Durante la elaboración del proyecto, se implementaron diferentes aspectos para garantizar el correcto funcionamiento del sistema, se realizaron sesiones de trabajo colaborativo en el cual se realizó un intercambio de ideas entre los integrantes de desarrollo para la elaboración del sistema, estas secciones fueron importantes, ya que permitieron identificar falencias en el uso de la aplicación y proponer ideas para solucionarlas en tiempo real, esto permitió realizar ajustes al desarrollo de las funcionalidades de la plataforma.

En la etapa de análisis se obtuvo información detallada sobre las necesidades de los estudiantes y la elaboración de los requisitos específicos de la plataforma, esta fase fue fundamental en la elaboración del proyecto pues permitió que los objetivos de aprendizaje y las experiencias de los usuarios finales se alinearan con las funcionalidades del sistema, lo que facilitó la implementación del diseño. La recopilación de los requisitos funcionales se realizó por medio de entrevistas, lo que permitió identificar y definir las características del proyecto, garantizando el propósito educativo del sistema adoptando las exigencias pedagógicas de la asignatura de métodos numéricos.

En la fase de construcción, se diseñaron los diagramas de clase, el cual permitió estructurar y organizar la arquitectura del sistema, al modelar las clases, con sus respectivos atributos y métodos se visualizó como interactúan los componentes dentro del proyecto, lo que facilitó la comprensión del sistema, además al proporcionar una base sólida para la implementación del sistema, los desarrolladores utilizaron el diagrama como guía para identificar las posibles mejoras que se pudieran realizar al proyecto.

El diagrama de casos de uso permitió estructurar y comprender las funcionalidades del sistema, permitiendo establecer diferencias entre las acciones que pueden realizar los actores involucrados dependiendo del rol de este mismo, el diagrama permitió visualizar las relaciones entre los casos de uso, como también las dependencias y las funcionalidades del sistema, además facilitó la identificación de los

requerimientos claves para el correcto funcionamiento del sistema cumpliendo con las necesidades del usuario.

El diagrama de despliegue identifica como los componentes del sistema de simulación están organizados y conectados en un entorno real, se detectaron los nodos principales y las herramientas empleadas en el diseño, permitiendo visualizar la integración de la funcionalidad y la eficiencia del sistema. El diagrama de despliegue desarrolla una perspectiva clara de las estructuras físicas y lógicas del sistema, identificando las conexiones y dependencias entre los nodos, lo cual permite comprender como fluye la información entre los componentes y como el sistema responde a las solicitudes del usuario. La implementación de estos diagramas en la fase de construcción aseguro la coherencia entre los componentes del sistema, la elaboración de los bocetos, facilito la organización grafica de los componentes que incorporan la plataforma y garantizar que el sistema sea atractivo y funcional para los estudiantes que quieran aprender sobre el método Runge Kutta.

En la fase de evaluación, se incorporaron ejercicios para identificar la usabilidad del sistema, con la participación de los docentes y estudiantes se realizó una retroalimentación exhaustiva, ya que proporcionaron comentarios claves para el mejoramiento de la plataforma, también se midió el impacto en aprendizaje del método Runge Kutta, los estudiantes mejoraron de manera significativa en la comprensión del método utilizando el sistema de simulación, este proceso destaco la eficiencia del uso de la plataforma, obteniendo resultados satisfactorios.

Cada una de las fases de la metodología XP que se implementó en el proyecto fue esencial y eficiente para desarrollar el Sistema de simulación dinámica de movimiento para el aprendizaje del método Runge-Kutta, permitiendo cumplir con los estándares técnicos y las expectativas de los usuarios.

Referencias Bibliográficas

- Arias, L. (2016). Lenguaje de modelamiento unificado (UML) para modelamiento de embotelladora. Revista Ingenierías Universidad Tecnológica de Pereira, 21(1).
<https://www.redalyc.org/pdf/849/84950584006.pdf>
- Arias, M. (2006). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *Revista de Ciencias de la Computación, 10, 1-17.
<https://repository.ucc.edu.co/server/api/core/bitstreams/a7ea175f-de13-49fd-afba-c021baf7a6d0/content>
- Beck, K., & Andres, C. (2004). Extreme programming explained: Embrace change (2^a ed.). Addison-Wesley.
[https://books.google.com.co/books?hl=es&lr=&id=G8EL4H4vf7UC&oi=fnd&pg=PR13&dq=Extreme+programming+explained:+Embrace+change+\(2%C2%AA+ed.\).+AddisonWesley.&ots=jcvFAvmYtl&sig=94tYG2vQaBzPEA772cP2IE1vnrc#v=onepage&q&f=false](https://books.google.com.co/books?hl=es&lr=&id=G8EL4H4vf7UC&oi=fnd&pg=PR13&dq=Extreme+programming+explained:+Embrace+change+(2%C2%AA+ed.).+AddisonWesley.&ots=jcvFAvmYtl&sig=94tYG2vQaBzPEA772cP2IE1vnrc#v=onepage&q&f=false)
- Cortez, J., Páez, J., & Simanca, F. (2021). Los diagramas de Unified Modeling Language (UML) para el mejoramiento de los requerimientos funcionales en el desarrollo de aplicaciones. Revista de Investigación en Tecnologías de la Información, 10(1).
https://www.researchgate.net/publication/361253874_Los_Diagramas_de_Unified_Modeling_Language_UML_para_el_Mejoramiento_de_los_Requerimientos_Funcionales_en_el_Desarrollo_de_Aplicaciones
- Díaz, L., Torruco, U., Martínez, M., & Varela, M. (2013). La entrevista, recurso flexible y dinámico. Investigación en Educación Médica, 2(7).
https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2007-50572013000300009
- Letelier, P., & Penadés, M. C. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).
https://www.researchgate.net/publication/26428496_Metodologias_agiles_para_el_desarrollo_de_software_eXtreme_Programming_XP
- Pérez, M., Irrazabal, E., Carrasco, R., & Coca, Y. (2016). Importancia de los requisitos no funcionales: estudio preliminar en una Universidad de Cuba.

- https://www.researchgate.net/publication/290996348_Importancia_de_los_requisitos_no_funcionales_estudio_preliminar_en_una_Universidad_de_Cuba
- Pradel, M., & Raya, M. (2010). Requisitos. Universitat Oberta de Catalunya. https://openaccess.uoc.edu/bitstream/10609/142846/5/PLA1_Requisitos.pdf
- Robles, D., Hernández, M., Mendoza, V., & Guaña, J. (2022). La educación tradicional vs la educación virtual. Recimundo, 6(4). <https://recimundo.com/index.php/es/article/view/1902>
- Sabaduche-Rosillo, D. (2015). Herramientas virtuales orientadas a la optimización del aprendizaje participativo: Estado del arte. Revista SME, 6(1). https://repositorio.usmp.edu.pe/bitstream/handle/20.500.12727/1703/sme_v6n1_art2.pdf
- Sayas, J., & García, J. (2014). Diccionario para la emisión de informes diagnósticos. https://repositorio.uci.cu/bitstream/ident/9156/2/TD_07790_14.pdf#page=56
- Tzenguzha-Abarca, F. X., Cárdenas-Cordero, N. M., Flores-Chuquimarca, D. K., & Brito-Albuja, J. G. (2021). Herramientas virtuales de aprendizaje para el desarrollo de la lectoescritura. Revista Arbitrada Interdisciplinaria Koinonía, 6(4), 351-375. <https://doi.org/10.35381/r.k.v6i4.1506>
- Velastegui, P. (2019). Plataformas virtuales y su impacto en la Educación Superior. Explorador Digital, 1(4), 5-21. <https://doi.org/10.33262/exploradordigital.v1i2.318>
- Vidal, C., Rivero, S., López, L., & Pereira, C. (2014). Propuesta y aplicación de diagramas de clases UML JPI. Información Tecnológica, 25(5). <http://dx.doi.org/10.4067/S0718-07642014000500016>
- Wang, X., Conboy, K., & Cawley, O. (2012). “Leagile” software development: An experience report analysis of the application of lean approaches in agile software development. Journal of Systems and Software, 85(6), 1287-1299. <https://doi.org/10.1016/j.jss.2012.01.061>