

CAPÍTULO 3. Entorno Interactivo en el Campus Virtual 3D “Udlaverso” para la Enseñanza de Programación Dinámica en Modelos Determinísticos

<https://doi.org/10.64325/5w6sp285>

Autores

Carlos Mauricio Roa Meneses

Universidad de la Amazonia

<https://orcid.org/0009-0003-6309-6126>

Fredy Antonio Verastegui Gonzalez

Universidad de la Amazonia

<https://orcid.org/0000-0002-0525-6879>

Lubeimar Eduardo Gallego Ruiz

Universidad de la Amazonia

<https://orcid.org/0000-0001-9117-6420>

Como citar: Roa Meneses, C. M., Verastegui Gonzalez, F. A., & Gallego Ruiz, L. E. (2025). Entorno Interactivo en el Campus Virtual 3D “Udlaverso” para la Enseñanza de Programación Dinámica en Modelos Determinísticos. En E. E. Millán Rojas (Ed.), *Uso de la realidad Virtual 3D para articular la enseñanza de la Física en Ingeniería de Sistemas* (pp. 75-101). Editorial Didaxis. <https://doi.org/10.64325/5w6sp285>

Resumen

En este capítulo se describe el desarrollo de una interacción dentro del entorno “Udlaverso”, un campus virtual 3D desarrollado por la Universidad de la Amazonia para la enseñanza de Programación Dinámica en Modelos Deterministas. El trabajo desarrollado se basa en la metodología Métrica Versión 3, del estándar de Ciclo de Vida de Desarrollo de Software ISO 12207, que estructura el proceso en tres fases: análisis, diseño y construcción. En esta fase se revisa el contenido temático impartido en la asignatura Modelos Determinísticos, y se construyen y se realizan las entrevistas al docente para establecer qué fue lo que les pareció más difícil a los estudiantes y qué sería necesario para llenar esos vacíos para una mejor comprensión del tema. La incorporación del simulador 3d para el proceso de enseñanza no es única en la instrucción de programación dinámica. En campos como la anatomía animal, los simuladores 3D también han demostrado ser un recurso útil para simplificar la asimilación y memorización de saberes. De acuerdo con la investigación sobre la utilización de simuladores 3D en anatomía, esta tecnología facilita a los alumnos la observación y manejo de modelos complejos,

favoreciendo una asimilación más efectiva de los conceptos mediante la experiencia práctica e inmersiva (Trujillo, 2019).

El resultado es un ambiente educativo que facilita a los alumnos la interacción práctica para la explicación de la programación dinámica. Los usuarios tienen la posibilidad de interactuar con el entorno y las variables y presenciar sus cambios de manera directa, lo que acelera el proceso de aprendizaje y fomenta una comprensión más fácil de los conceptos. El capítulo concluye destacando el impacto beneficioso de este ambiente en el proceso educativo, mostrando un instrumento revolucionario para potenciar el aprendizaje en el campo espinoso de la programación dinámica.

Palabras claves: Programación dinámica; Entornos virtuales de aprendizaje; Simulación tridimensional; Enseñanza de la informática

Abstract

This chapter describes the development of an interaction within the “Udlaverso” environment, a 3D virtual campus developed by the University of the Amazon for teaching Dynamic Programming in Deterministic Models. The work carried out is based on the Metric Version 3 methodology of the ISO 12207 Software Development Life Cycle standard, which structures the process into three phases: analysis, design, and construction. In this phase, the subject matter taught in the Deterministic Models course is reviewed, and interviews are conducted with the instructor to determine what students found most difficult and what would be necessary to fill those gaps for a better understanding of the subject. The incorporation of 3D simulators into the teaching process is not unique to dynamic programming instruction. In fields such as animal anatomy, 3D simulators have also proven to be a useful resource for simplifying the assimilation and memorization of knowledge. According to research on the use of 3D simulators in anatomy, this technology makes it easier for students to observe and handle complex models, promoting more effective assimilation of concepts through practical and immersive experience (Trujillo, 2019). The result is an educational environment that facilitates practical interaction for students to explain dynamic programming. Users have the opportunity to interact with the environment and variables and witness their changes directly, which accelerates the learning process and promotes easier understanding of concepts. The chapter concludes by highlighting the beneficial impact of this environment on the educational process, showcasing a revolutionary tool for enhancing learning in the challenging field of dynamic programming.

Keywords: Dynamic programming; Virtual learning environments; Three-dimensional simulation; Computer science education

Introducción

En el capítulo se presenta un entorno virtual 3d para el aprendizaje de la asignatura de modelos determinísticos de la carrera de Ingeniería de sistemas de la Universidad de la Amazonia, el cual el sistema se presenta como un producto de innovación enfocado al aprendizaje didáctico de temas complejos de programación dinámica, como el problema de la mochila y la ruta más corta, el cual se ve la escasez de material educativo eficiente e innovador para el aprendizaje de este, para eso este

entorno interactivo busca solventar esa necesidad, dando un sistema inmersivo el cual permite llevar estos temas complejos y acoger estos conceptos abstractos y llévalos a una simulación de la vida real.

En el contexto actual, la enseñanza de la programación dinámica es compleja, debido a que se requiere que el profesor lleve al estudiante de manera práctica a través de un proceso de comprensión y aplicación de sus principios en problemas prácticos, lo cual más desafiante debido a la ausencia de herramientas interactivas que permitan a los estudiantes ver por sí mismos cuáles son los efectos de sus decisiones al momento de resolver problemas de y poder llevar esos ejercicios a la realidad es muy poder simular o recrear esto. A partir del problema, surge la necesidad de un entorno interactivo educativo 3D que no solo represente conceptos teóricos, sino que también brinde la oportunidad de experimentar con el entorno de la temática, y trabajar de forma inmersiva y de manera dinámica los cuales ayuden al aprendizaje de los estudiantes de Ingeniería de Sistemas. Se ha empleado exitosamente escenarios para la explicación de la temática de programación dinámica, incluyendo el problema de la mochila y la optimización de rutas más corta. Un caso significativo es la aplicación de la programación dinámica determinística (PDD) para determinar el camino ideal entre nodos tal como se muestra en la investigación de (Acosta, 2019).

Se emplearon algunas como la metodología Métrica V3, fundamentada en la norma ISO 12207 para el ciclo de vida del software. Se optó por esta metodología ya que proporcionar una estructura más del proyecto con las etapas de análisis, diseño y edificación, asegurando de esta manera un proceso ordenado y regulado. De esta manera se trabajó bajo el principio que plantea sobre el análisis de sistema la cual fue la base para construir y estructurar los requisitos funcionales, abstrayendo los materiales y métodos avanzados, para elaborar a partir de la herramienta Visual Paradim para modelar los casos de uso, diagrama de clases, diagrama de despliegue, y utilizando el visor 3D FireStorm, que permite construir una experiencia visual y práctica en 3D. Se uso de apoyo la inteligencia artificial (AI), para la creación de los elementos interactivos dentro del “UdlaVerso” y para construir experiencias inmersivas, para construir scripts que ayudan a enriquecer la enseñanza inmersiva lo que permitió

adaptar las interacciones a las necesidades principios de enseñanza de programación dinámica en modelos determinísticos (Peña, 2020).

Una vez obtenidos los resultados se muestra como está enseñanza alterna a la tradicional y al interactuar con estos conceptos en un ambiente inmersivo en los cuales simulamos estas realidades distintas se a los usuarios puede poner en un mejor contexto para aprender estos temas de matemáticas más complejos en la facilidad y el mejor entendimiento a disposición de los estudiantes y personas interesadas supliendo la necesidad de herramientas alternas para la integración de aprendizajes, el cual con un poco de creatividad, atractivo e innovador y curioso para quien quiere aprender en un entorno diferente.

Materiales y Métodos

Para este capítulo se basó la metodología por la métrica v3 el cual es un modelo que se creó para llevar un control del proceso completo de la vida del desarrollo de un software definido por la norma ISO 12.207 “Tecnologías de la Información - Procesos del Ciclo de Vida del Software”. Del cual se tomaron partes de esta metodología para abarcar este capítulo: Análisis de Sistemas de Información (ASI), Diseño de Sistemas de Información (DSI) y Construcción de Sistemas de Información (CSI) (Salamanca Aguado, 2020).

Fase de análisis del Sistema de Información (ASI)

Para el primer proceso de ASI, se investigó en el programa de Ingeniería de Sistemas de la Universidad de la Amazonia el pensum detallado de los ejes temáticos de la asignatura modelos determinísticos, posteriormente se realizó determinó la unidad a trabajar, y luego se trabajó sobre ella que es la “Unidad 4. Programación Dinámica” se consultó sobre esa temática, para dar un enfoque, y a partir de lo consultado se intentó abordar las preguntas en relación, y se construyó una entrevista para el docente de esta asignatura, la cual fue en ideal la entrevista, ya que la intención era captar una idea general para a partir de allí desglosar las dificultades y sobre ello construir las necesidades identificando los elementos más importantes y puntuales de cada temática, así contando cómo a partir de su experiencia, que dificultades ha tenido a la hora de

enseñar y cómo podemos mejorar y enfrentar los desafíos que tienen los estudiantes los y como mediante nuestro sistema de entorno virtual solventar ese problema

Por ende, se presenta la estructura preguntas para entrevista las cuales son el resumen de la entrevista realizada:

Tabla 1. Preguntas de la entrevista.

N°	Pregunta
1	¿Podría proporcionar una visión general y detallada de cómo percibe la experiencia de una clase en el UdlaVerso?
2	¿Cómo enseña actualmente conceptos como programación dinámica y distribución de recursos?
3	¿Qué funcionalidad específica que cree que debería existir?
4	¿Podría darme algún ejemplo de cómo sería esa interacción?
5	¿Qué tipo de ejemplos o aplicaciones prácticas cree que podrían ayudar a los estudiantes a entender mejor estos conceptos?
6	En cuanto a la evaluación, ¿qué métodos utiliza para medir la comprensión de los estudiantes sobre programación dinámica y distribución de recursos?
7	¿Cómo imagina que los estudiantes interactúen con el software? ¿Qué funcionalidades serían esenciales para facilitar el aprendizaje?
8	¿qué espera que los estudiantes puedan hacer y aprender con este problema de rutas más cortas y largas?
9	¿Considera importante que el sistema proporcione explicaciones detalladas y justificaciones en cada paso del proceso para estos tres temas?

Nota. Esta tabla muestra las preguntas para la entrevista al docente de la asignatura de Modelos determinísticos. Fuente: elaboración propia.

Fase de Diseño del Sistema de Información (DSI)

Para está fase teniendo en cuenta la métrica v3 se realizó la aplicación de lo recolectado en la fase de análisis y se pasó al paradigma de UML (Unified Modeling Language) el cual busca poder modelar, visualizar, especificar, construir y documentar partes de un sistema software (García-Peñalvo, 2021). Haciendo uso de la herramienta online en la versión web llamada Visual Paradim a partir de lo información trabajada, se crearían el caso de uso, con sus respectivos actores y relaciones con las acciones o funcionalidades.

Por consiguiente, se construyó del diagrama de despliegue, donde se ilustra la arquitectura y distribución de los recursos del sistema, los componente y equipos físicos,

además de las conexiones entre estos elementos, y definir el cómo está implementada la infraestructura necesaria para el correcto funcionamiento del sistema “UdlaVerso”.

Fase de Construcción del Sistema de Información (CSI)

Para esta fase, se realizó un prototipo manual, el cual su fin era dar una idea básica y rápida del escenario generar, definir sus escenario y modelo de la infraestructura que iría en el UdlaVerso para evaluar y decidir si era o no factible para las siguientes etapas. Seguidamente se crearon bocetos utilizando GPT-4 y Meta AI. Para recrearlo un poco más realista y ser un poco más preciso. Por consiguiente, se realizaría la instalación y despliegue del visor FireStorm, plataforma utilizada para la visualización y el desarrollo de los elementos del sistema, el cual permite construir los objetos darle las funcionalidades y recrear lo que necesitamos a partir de los recolectado en las fases anteriores. Consecuentemente se construiría los elementos más detallados y forma meticulosa, teniendo en cuenta los requerimientos y lo establecido por el diseño realizado, para ello se crearon los modelos 3D, se le aplicaron texturas y se crearon las funcionalidades específico para los elementos que necesitaras realizar acciones dentro del sistema. Para la creación de estos scripts se utilizarían GPT-4 para lograr un código eficiente en el lenguaje LSL. Para crear lo más preciso posible a lo que se requería construir en el tema de funcionalidades a los objetos.

Además, se llevaría un control en Excel un listado detallado de los canales de comunicación entre objetos, el cual llevaba un control para no tener problemas con los distintos objetos e interferir entre ellos y causar conflictos o entorpecer las funcionalidades por estar usando dos objetos el mismo canal

Para la etapa final de la fase de construcción, la creación de una rúbrica de evaluación es fundamental. Según (Andrade, 2005), las rúbricas facilitan y dan una guía estructurada y precisa para comprender cuales son las expectativas, y al mismo tiempo, permitir a los docentes evaluar el trabajo de manera objetiva, justa y precisa con unos parámetros bien definidos (Andrade, 2005). Para esta rúbrica se evalúa distintos criterios los cuales se detallan a continuación: Visibilidad del estado de sistema; Correspondencia entre el sistema y la realidad; Libertad y control por parte del usuario;

Que haya consistencia y estándares; Prevención de errores; Reconocimiento antes que recuerdo; Flexibilidad y eficiencia en el uso; Diálogos estéticos y diseño minimalista; El sistema ayuda a los usuarios a reconocer, diagnosticar y recuperar los errores y El sistema tiene ayuda y documentación. Para este contexto se evaluó cada escenario del “Entorno Interactivo En El “Udlaverso” Para La Enseñanza De Programación Dinámica En Modelos Determinísticos”.

Cómo herramienta para esta evaluación se construyó un formulario en Google Forms ubicando cada escenario de la aplicación y evaluando cada principio de Nilesen en una escala valorativa de uno (1) a cinco (5), donde uno es insuficiente y cinco es sobresaliente, los resultados se analizaron de acuerdo con cada escenario y en conjunto con el total de las evaluaciones realizadas por el grupo de estudio.

Resultados

Dentro de los resultados obtenidos se observó que la enseñanza de Programación Dinámica y modelos determinísticos, se destaca como los estudiantes pueden interactuar de manera inmersiva, practica y visuales; conceptos complejos como la programación dinámica, el problema de la mochila y la ruta más corta, ya que dentro de estos entornos les permite equivocarse cambiar variables, y recibir visualmente la soluciones mejores y la explicación de estos conceptos internamente lo cual es crucial para la comprensión de quienes interactúen en este mundo de simulación de la realidad.

Dentro del estudio realizado se encontraron los postulados de Barahona Henao (2015), del cual aplico en su investigación una simulación inmersiva en 3D para la enseñanza de física y permitir a sus estudiantes interactuar con sus parámetros y ver cómo afecta en los resultados el cambio de los valores iniciales, en tiempo real con estos cambios en su investigación menciona que: “La simulación expresiva es un instrumento que facilita la materialización y comprensión de los fenómenos físicos. permite la interacción con los estudiantes a través del teclado, permitiéndole cambiar algunos parámetros tales como: la masa del cuerpo, el coeficiente de rozamiento y visualizar las instrucciones claves para el manejo de la simulación y la puesta en marcha del suceso físico e identificar los elementos que intervienen en él” (Y, 2015). De manera

similar el UdlaVerso trata de simular la realidad y potenciar contexto que quizás en la realidad se podría ser más complejos recrear, por ende, a lo mismo que Barahona, se utiliza la simulación 3D para interactuar los usuarios y a partir de datos de entrada modificar estos y ver como repercute en el resultado, viéndolo de manera instantánea y dinámica.

Resultados de la Fase de Análisis del Sistema de Información (ASI)

Se revisó el pensum del programa de ingeniería de sistemas de la Universidad de la Amazonia, el cual serviría para revisar los contenidos temáticos de la asignatura de Modelos Determinísticos, la cual se usó la unidad 4. Que corresponde a la programación dinámica bajo, los cuales temas como el problema de la mochila y la ruta más corta, los cuales fueron tomados como fundamentos teóricos y así orientar la estructura de la construcción de la entrevista que se llevó a cabo. Se elaboró un conjunto de preguntas, orientadas a la temática determinada la cual a partir de los conceptos se buscaba recolectar la información valiosa necesaria para ver la dificultades y problemáticas a la hora de la enseñanza de la temática de programación dinámica, para esto se orientó la entrevista al docente experto de la asignatura.

Tabla 1. *Respuestas a la entrevista realizada*

N°	Respuesta
1	Me gustaría una manera que ayude a estudiantes a comprender conceptos complejos en áreas como el problema de la mochila, ruta más corta, y la programación dinámica. La idea es que sea de manera práctica y visualmente atractiva que facilite el aprendizaje y la enseñanza
2	Claro, actualmente utilizo una combinación de clases teóricas y ejercicios prácticos en papel. Sin embargo, he notado que a los estudiantes les cuesta visualizar y aplicar estos conceptos en situaciones reales, lo que puede hacer que pierdan interés
3	Bueno, en incorporar módulos donde los usuarios puedan interactuar directamente con conceptos clave, experimentando y viendo resultados en tiempo real
4	Por ejemplo, en el caso de cada problema, los usuarios podrían manipular ciertos elementos y observar cómo afectan a un resultado o solución
5	Creo que ejemplos interactivos donde puedan manipular variables y ver resultados en tiempo real serían muy útiles. Por ejemplo, problemas como el de la mochila, donde puedan experimentar con la selección de objetos y ver cómo afecta al resultado óptimo
6	Utilizamos exámenes escritos, ejercicios prácticos y proyectos

7	Me gustaría que el software permita a los estudiantes seleccionar y modificar objetos con diferentes pesos y valores de una mochila virtual. Es esencial que puedan ver el proceso de solución paso a paso, quizás a través de una tabla o gráfica que muestre cómo se calcula el valor óptimo en cada iteración del algoritmo de programación dinámica. Además, sería útil que el software explique las decisiones tomadas en cada paso
8	Quiero que los estudiantes puedan construir ir añadiendo nodos y conectándolo que representen distancias. Debería permitirles calcular tanto la ruta más corta entre dos nodos seleccionados, mostrando el paso a paso de algoritmos como para la ruta más. Es importante que puedan visualizar cómo se exploran los caminos y cómo se actualizan las distancias en cada paso
9	Me gustaría que las explicaciones detalladas son cruciales para que los estudiantes comprendan por qué, que hay detrás de cada paso. Esto les ayudará a internalizar los conceptos y a aplicarlos de manera autónoma en diferentes situaciones

Nota. En esta tabla se muestra la respuesta a la entrevista dirigida al docente de la Asignatura de los Modelos Determinísticos. Fuente: elaboración propia.

Para esta fase se orientó en base del trabajo de (Ceballos, 2021), para la selección y aplicación de la técnica en el levantamiento de requerimientos. Este artículo permitió un marco sistemático y organizado para la identificación de las necesidades específicas del usuario final esto estructuró en un posterior análisis en un enfoque en la necesidades y experiencias de usuario. El método propuesto en esta investigación permitió una mejor definición de los requerimientos y proporcionó herramientas de recolección de información efectivas para el entendimiento de las necesidades del usuario y el desarrollo del sistema, aspectos críticos y fundamentales en la fase inicial del proyecto.

Resultados de la Fase de Diseño

Durante la fase posterior a la realización de la entrevista y la transferencia de esta información al docente, se buscó recopilar los datos necesarios para definir los requerimientos funcionales, los casos de uso y sus especificaciones. Esta información fue empleada en la construcción del diagrama de casos de uso, con el objetivo de identificar claramente cada caso y los actores involucrados. A continuación, y considerando que "esta forma de ver a los datos como objetos activos que interactúan entre sí y que son capaces de reaccionar y cambiar su estado interno en función de los mensajes que reciben, da lugar a todo un nuevo paradigma de programación llamado

orientación a objetos o Programación Orientada a Objetos (POO), La Programación Orientada a Objetos es un paradigma de programación que concibe a los programas como formados por entidades llamadas "objetos". Estos objetos son capaces de recordar su propio estado interno y comunicarse entre sí mediante el paso de mensajes. Dicho intercambio permite a los objetos cambiar sus estados, compartir información y solicitar procesamiento a otros objetos. Los objetos tienen atributos que almacenan valores y métodos que definen su comportamiento, lo que permite que reaccionen a mensajes y modifiquen su estado en consecuencia. Esta forma de estructurar los datos, ocultando sus detalles y exponiendo únicamente las operaciones que manipulan esos datos, define la esencia del paradigma orientado a objetos (López, 2018), tal como se describe en el material del cual se utilizó este modelo para construir el diagrama de clases, seguido del diagrama de despliegue con la herramienta Visual Paradigm.

Tabla 2. *Requerimientos Funcionales.*

Identificador	Nombre	Requisito
RF01	Interactuar con la Simulación del Problema de la Mochila	El sistema debe permitir a los estudiantes interactuar con una simulación del problema de la mochila. Los usuarios podrán seleccionar y asignar recursos a una mochila virtual, para maximizar la ganancia total sin exceder el peso permitido. Además, se proporcionarán botones para agregar peso y puntos de ganancia, calcular la solución óptima y visualizar, en tiempo real, cómo se construye la solución paso a paso.
RF02	Desarrollar un espacio para la explicación del tema de la mochila	El sistema debe proporcionar un entorno interactivo donde los estudiantes puedan explorar y comprender el problema de la mochila. Este espacio incluirá una explicación detallada sobre el problema, definición de que es, conceptos, y ejemplos

RF03	Construir una simulación de Rutas Más Cortas	El sistema debe permitir a los estudiantes crear y personalizar nodos de conexión en una simulación interactiva. Los usuarios podrán diseñar añadiendo nodos y ajustando parámetros como el peso de las aristas y la posición de los nodos. El objetivo es observar cómo estos factores impactan en las rutas más cortas. La simulación debe mostrar los caminos posibles y ofrecer explicaciones detalladas sobre dichas rutas.
RF04	Desarrollar un espacio para la explicación de las rutas más cortas	El sistema debe ofrecer un entorno interactivo en donde los estudiantes puedan explorar y comprender las rutas más cortas y largas dentro de una red de internet en donde se vea la teoría de la ruta más corta en programación dinámica
RF06	Desarrollar un espacio de evaluación interactiva del Conocimiento del	El sistema debe ofrecer a los estudiantes un formulario de evaluación diseñado para medir el conocimiento adquirido sobre los problemas abordados, como el problema de la mochila y las rutas de más corta
RF07	Desarrollar un espacio de bienvenida y e introducción al sistema	Desarrollar un espacio de bienvenida e introducción al sistema. El sistema debe ofrecer a los usuarios una página inicial donde se les dé la bienvenida y se aprenda a manejar el entorno. Este espacio debe ser atractivo e intuitivo, facilitando la comprensión y navegación del sistema para los usuarios nuevos. Además, se pueden incluir guías que ayuden en la familiarización con las funcionalidades disponibles

Nota. En esta tabla se muestra la construcción de los requerimientos funcionales a partir de la información recolectada. Fuente: elaboración propia.

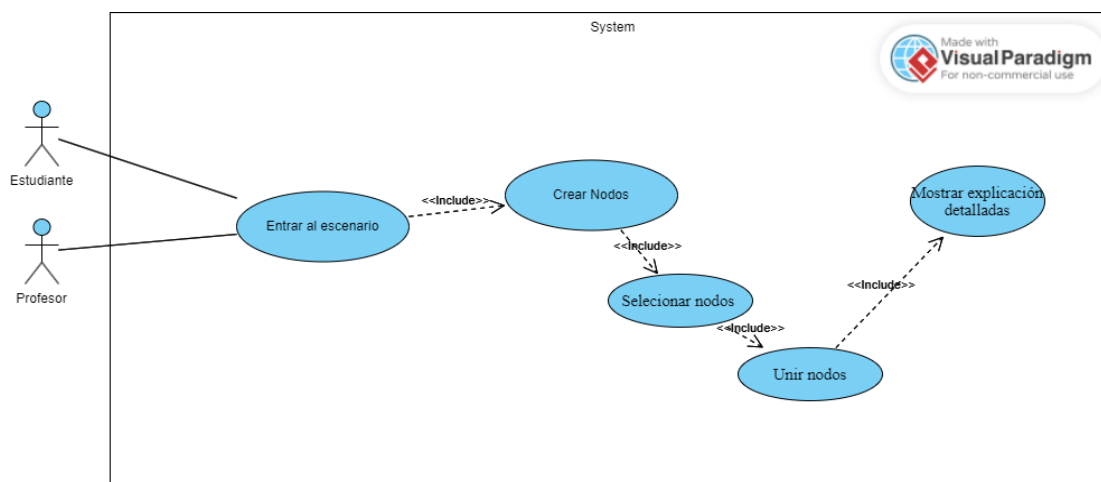
Tabla 3. Especificación de requisito funcional RF01

Especificación de requerimiento	
Código	ERF02
Nombre	Modificar Objetos Seleccionados
Descripción	El sistema debe permitir a los usuarios modificar el peso y/o el valor de los objetos previamente seleccionados en la mochila virtual. Los cambios realizados deben reflejarse de manera inmediata, actualizando el peso total y el valor acumulado en la mochila
Prioridad	Alta
Origen	Pregunta 7
Usuarios	Profesor, Estudiante.

Nota. En esta tabla se muestra la especificación del requerimiento funcional RF01. Fuente: elaboración propia

A continuación, se presenta el resumen del análisis realizado a partir del diagrama de Casos de uso, el cual define los procesos y subproceso que se van a trabajar en el software, todo esto bajo el criterio y las observaciones del usuario.

Figura 1. Caso de uso para el requerimiento funcional RF03



Nota. Elaboración de casos de uso del RF03 a través de Visual Paradigm. Fuente: elaboración propia. Se elaboró usando cómo base (Burgués, 2018).

Tabla 4. Especificación de los casos de uso

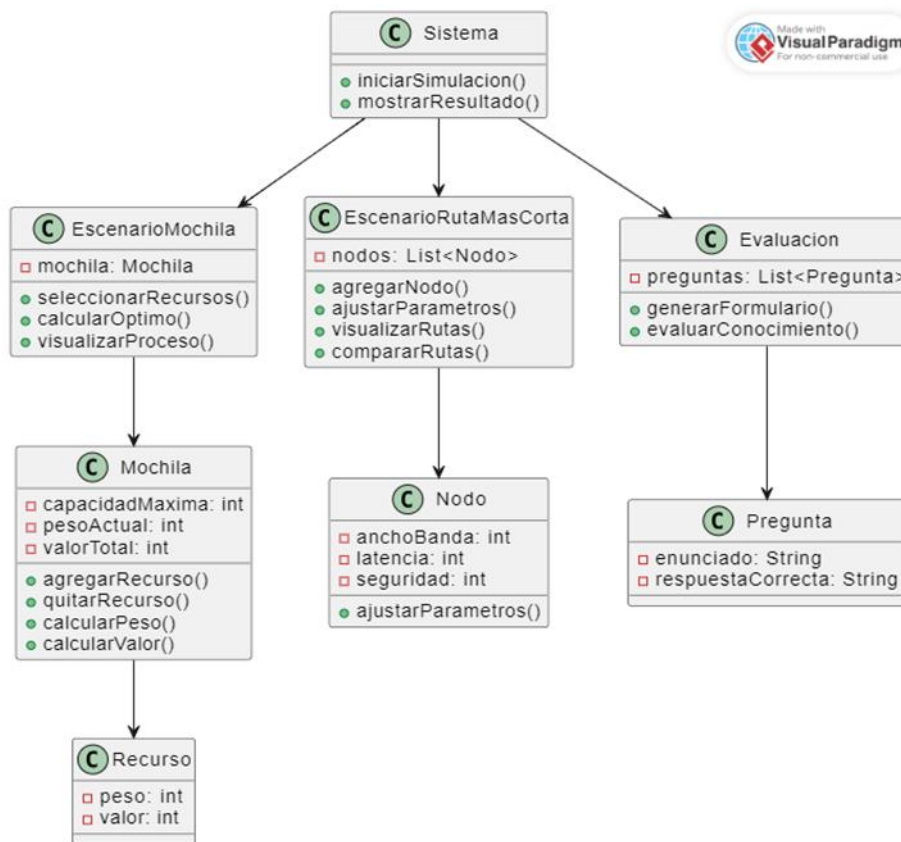
CU-<001>	Mostrar los objetos	
Versión	1.0	
Autores	Carlos Roa	
Fuentes	RF01	
Actores:	Profesor, Estudiante	
Objetivos Asociados	Permitir la selección de objetos con diferentes pesos y valores para agregar a la mochila virtual	
	Facilitar la gestión de objetos por parte del usuario	
Casos de uso Asociados	CU-003: Modificar objetos seleccionados.	
	CU-004: Eliminar objetos de la mochila	
Descripción	Este caso de uso permite a los usuarios (profesor y estudiante) seleccionar objetos con diferentes atributos (peso y valor) y agregarlos a una mochila virtual para simular el problema de la mochila. Los usuarios también pueden eliminar objetos de la mochila y visualizar el impacto de sus decisiones.	
Precondición	El usuario debe estar autenticado en el sistema.	
	El sistema debe mostrar un listado de objetos con sus atributos (peso y valor) disponibles para seleccionar.	
Atributos	Peso: El peso de cada objeto varía y está determinado previamente en el sistema.	
	Valor: El valor de cada objeto también está predeterminado y visible para el usuario.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la interfaz de selección de objetos en la mochila virtual.
	2	El sistema muestra un listado de objetos con sus respectivos pesos y valores.
	3	El usuario selecciona uno o varios objetos para agregar a la mochila.
	4	El sistema actualiza la mochila virtual mostrando los objetos seleccionados.
	5	El usuario puede visualizar el peso total y el valor acumulado en la mochila.
	6	El usuario tiene la opción de eliminar objetos seleccionados de la mochila.
Postcondición	El sistema refleja correctamente la mochila con los objetos seleccionados o eliminados.	

	Se debe probar que la funcionalidad de agregar y remover objetos opere sin errores.	
Excepciones	Paso	Acción
	3	Si un objeto no puede ser agregado (debido a restricciones del sistema o límite de capacidad), el sistema debe mostrar un mensaje de error indicando la causa.
	6	Si el usuario intenta remover un objeto que ya no está disponible, el sistema debe notificarlo y actualizar la visualización.
Rendimiento	3	El sistema debe procesar la selección de objetos en un máximo de 2 segundos.
	6	La eliminación de objetos debe completarse en un máximo de 2 segundos.
Importancia	Alta	
Urgencia	Alta	
Estado	En desarrollo	
Estabilidad	Alta	
Comentarios	Este caso de uso es fundamental para la simulación del problema de la mochila.	
	Se debe realizar una validación adicional para evitar que se seleccionen objetos que sobrepasen la capacidad de la mochila.	

Nota. Elaboración de Se Especificación de Caso de uso del requisito funcional RF01 a través de Visual Paradigm. Fuente: elaboración propia.

Para el diagrama de clases, se realizó siguiendo los conceptos desarrollados en el artículo “Cómo hacer el diagrama de clases” (Gama E, SF). Este artículo fue esencial para definir la estructura de clases del sistema, ya que permitió emplear un enfoque orientado a objetos en el que primero identificamos las clases principales junto con sus atributos y métodos correspondientes.

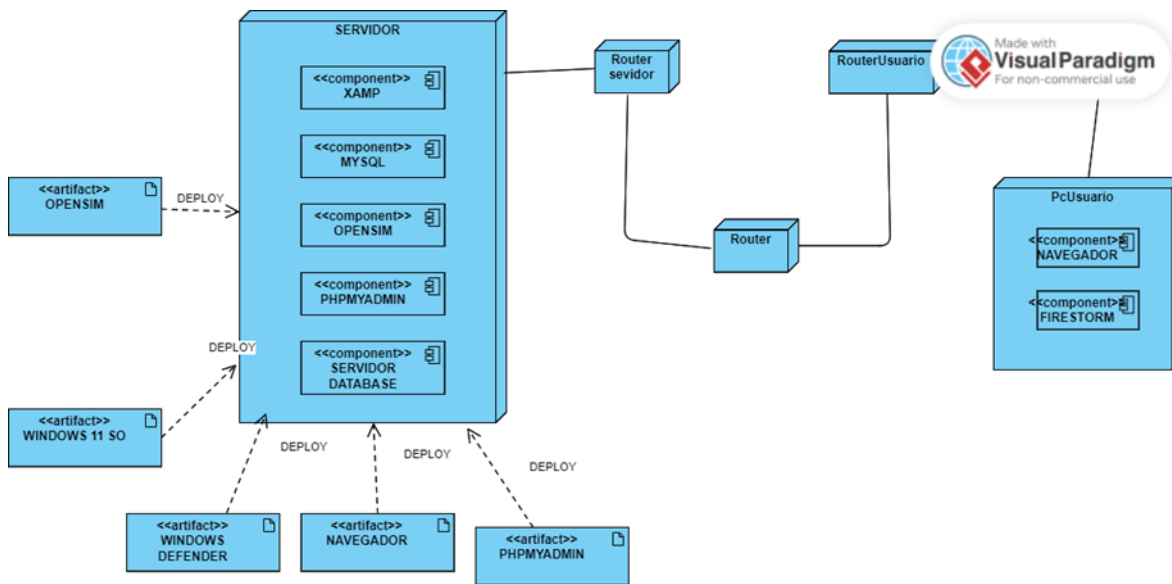
Figura 2. Diagrama de clases



Nota: Diagrama de clases elaborado con Visual Paradigm. Este diagrama representa la estructura de clases y sus relaciones para un sistema que aborda simulaciones y evaluaciones. Incluye componentes clave para resolver problemas como el de la mochila y rutas más cortas, además de integrar una funcionalidad de evaluación. Fuente: elaboración propia.

El diagrama se comenzó con la elaboración de la abstracción y el encapsulamiento de cada objeto, la cual fue asignado durante el análisis hasta llevar a convertirse a clase para después identificar de manera más fácil las relaciones y las dependencias entre ellas. Este enfoque ayudó a estructurar un modelo visual claro y coherente, para su correcto funcionamiento en el sistema. A partir de la arquitectura de Second Life, se construyó el diagrama de despliegue que permite comprender la estructura técnica que soporta su funcionamiento. Este diagrama muestra la infraestructura de red y los elementos necesarios para una experiencia inmersiva, así como la distribución de recursos de hardware y software

Figura 3. Diagrama de Despliegue



Nota. Diagrama de despliegue elaborado con Visual Paradigm. Este diagrama muestra la arquitectura de implementación de un sistema que incluye componentes principales como servidores, software y herramientas necesarias para el funcionamiento de OpenSim y servicios relacionados. Fuente: elaboración propia.

El artículo, Desarrollo Orientado a Objetos con UML de Ferré Grau y Sánchez Segura, se utilizó como base para construir el diagrama de despliegue de este capítulo. Este documento presenta los elementos y técnicas básicos para crear diagramas de implementación precisos en UML para modelar la infraestructura física de un sistema. El proceso comenzó identificando nodos y componentes críticos.

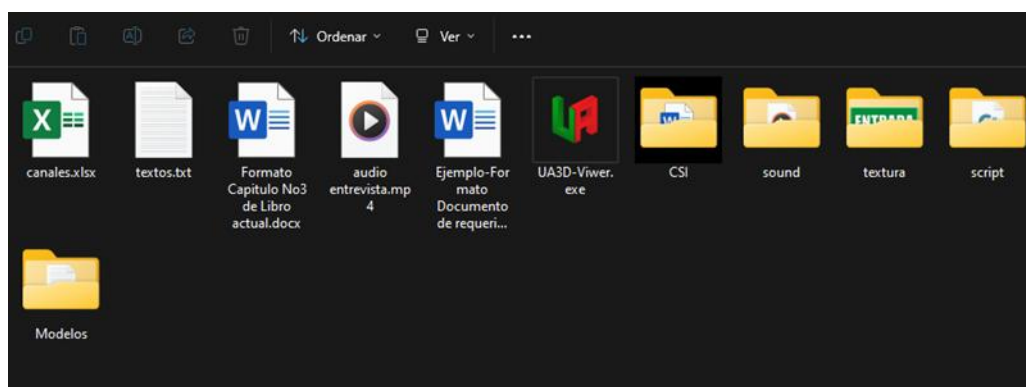
Siguiendo las recomendaciones de este artículo, cada nodo, incluidos servidores, procesadores y periféricos, se muestra gráficamente y está etiquetado correctamente. Luego, con referencia del material para colocar componentes de software en sus nodos correspondientes, asignar funcionalidades específicas y definir dependencias. Finalmente, de acuerdo con las técnicas UML descritas en este artículo, el diagrama se ha modificado utilizando técnicas de modelado comunes, como la arquitectura de tres niveles, para promover una estructura modular y organizada.

Este enfoque detallado aseguró que el diagrama cumpliera con los requisitos técnicos y representara claramente la implementación física del sistema dentro del entorno (Grau, 2008).

Resultados de la Fase de Construcción del Sistema de Información (CSI)

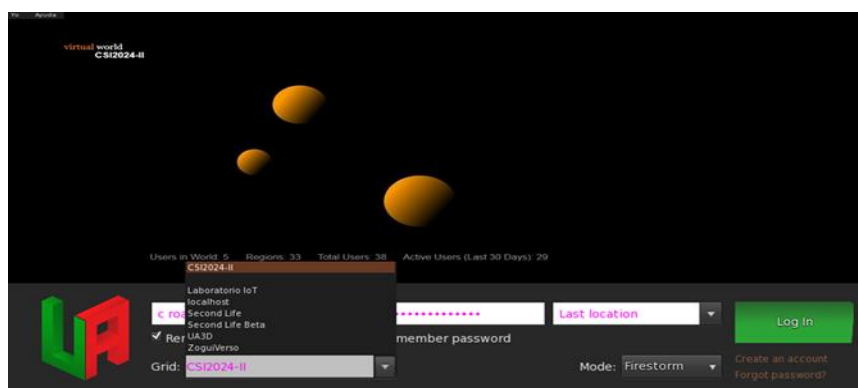
Una vez finalizada la fase anterior, el primer paso es descargar, configurar e ingresar al visor FireStorm, esta herramienta que permite gestión del entorno virtual en la que se puede visualizar y manipular los objetos en 3D dentro del Udlaverso. Una vez iniciado y con las credenciales de acceso correctamente configuradas, se inicia la construcción en el Udlaverso utilizando los modelos 3D generados por IA, integrándolos y ajustándolos en el entorno para dar forma al espacio virtual. Además, se emplea IA para generar los scripts en LSL que controlan el comportamiento de los objetos y un archivo Excel para gestionar los canales de comunicación, optimizando así la interacción entre elementos dentro del espacio virtual.

Figura 4. Descargar e instalar visor 3d FireStorm



Nota. Se puede observar el ejecutable UA3D-Viwer.exe. el cual se utiliza para entrar al sistema Udlaverso. Fuente: elaboración propia.

Figura 5. Configuración y conexión del visor 3d FireStorm



Nota. Se puede observar la conexión con la ip y puerto asignado se digita para configurar y habilitar la conexión con el servidor del Udlaverso. Fuente: elaboración propia.

A continuación, se presentan los bocetos de los diferentes escenarios creados con ayuda de Inteligencia artificial, estos son simplemente modelos de referencia para luego empezar la construcción con las prim en el entorno virtual.

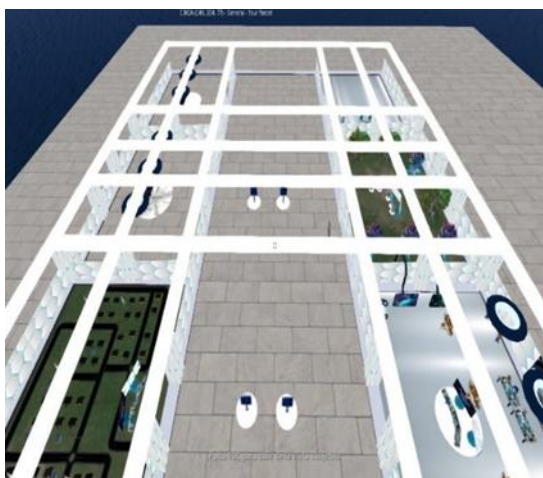
Figura 6. *Boceto Inicial IA*



Nota. En la ilustración boceto generado con IA para escenario general. Fuente: elaboración propia modificado con Meta AI.

En las siguientes imágenes se presenta el desarrollo de cada escenario construido en el entorno virtual 3D a partir de los requerimientos y el diseño de la aplicación.

Figura 7. *Escenario general Udlaverso*



Nota. En la ilustración escenario general Udlaverso. Fuente: elaboración propia.

En la siguiente figura se ilustrar un ejemplo de la programación realizada en un objeto del ambiente virtual, es de aclarar que existen múltiples scripts como este en el proyecto. Aquí se presenta uno de ellos como ejemplo demostrativo del trabajo realizado.

Figura 8. *Script del problema de la mochila*

```

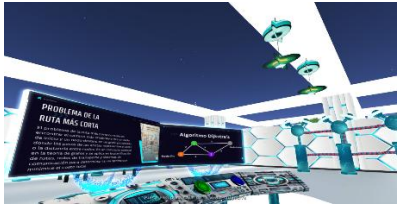
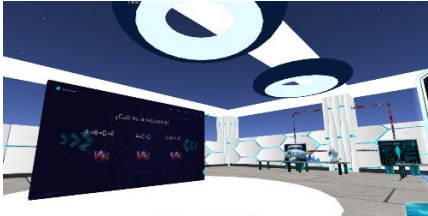
Script New Script
Archivo  Editar  Ayuda
50
51  listen(integer channel, string name, key id, string
message) {
52      if (channel == dialogChannel) {
53          if (message == "Aum. 10 kg") {
54              adjustWeight(10.0);
55          } else if (message == "Red. 10 kg") {
56              adjustWeight(-10.0);
57          } else if (message == "Aum. 10p") {
58              adjustGain(10.0);
59          } else if (message == "Red. 10p") {
60              adjustGain(-10.0);
61          }
62      }
63      |
64      llListenRemove(dialogChannel);
65      llSetTimerEvent(0.0);
66  }
67  }
68
69  start() {

```

Nota. Este script ajusta el peso o ganancia del objeto en el escenario del problema de la mochila, según el comando específico en el canal “dialogChannel” y luego finaliza la escucha para evitar seguir procesando mensajes innecesarios. Fuente: elaboración propia. En la figura 3.9 se presentan todos los escenarios construidos en la aplicación para abordar los requerimientos planteados por los usuarios, en ellos se encuentran escenarios de práctica, teóricos y de evaluación.

Figura 9. *Visualización general del entorno y sus diferentes escenarios*

Escen arios	Imagen	Descripción
1		Escenario teórico e introducción a la programación dinámica y el problema de la mochila

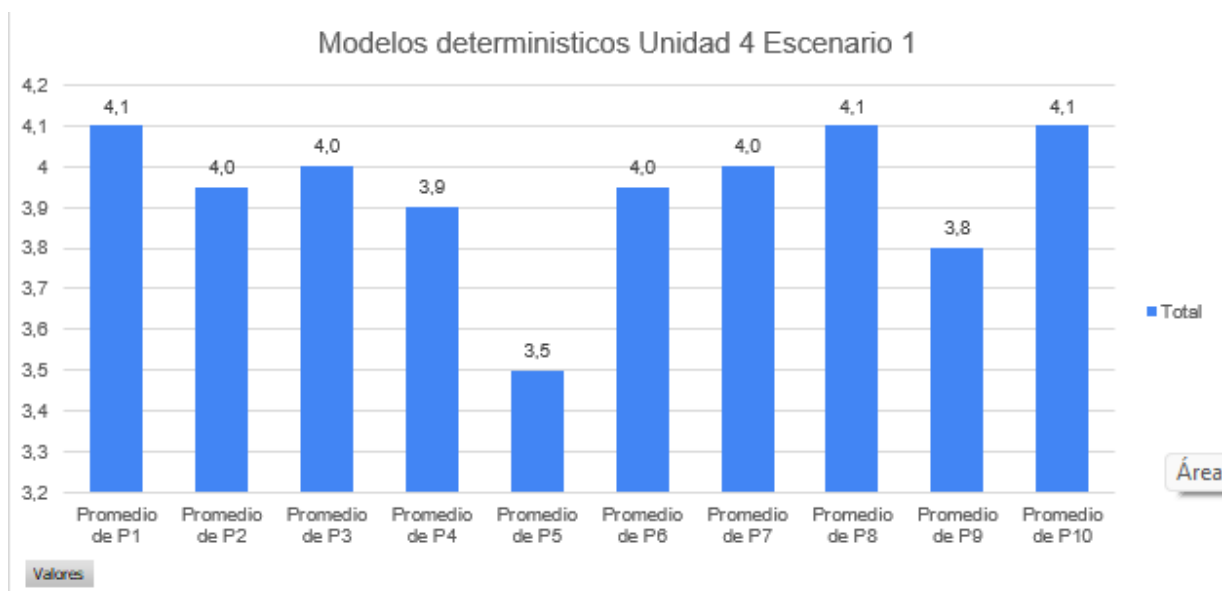
2		Escenario práctico e interactivo sobre el problema de la mochila en el cual se puede definir en los objetos los pesos y la ganancia y la capacidad de la mochila
3		Escenario teórico y explicativo sobre el problema de la ruta más corta y cómo se comporta el algoritmo internamente
4		Escenario práctico en el cual interactúas y creas dos conectarlos y darle peso a las alistas y se busca la ruta más corta
5		Escenario evaluativo que por uso de pares en la cual cada pregunta tiene una respuesta correcta que debe ser emparejada o relacionada
6		Escenario evaluativo que con un problema planteado de la ruta más corta debe encontrar la ruta más corta y elegir la respuesta correcta
7		Escenario del lobby el cual da la bienvenida e introduce en el mundo del UdlaVerso

Nota. En esta tabla se puede visualizar los 7 escenarios dentro del UdlaVerso y la descripción de cada uno. Fuente: elaboración propia.

Evaluación Heurística

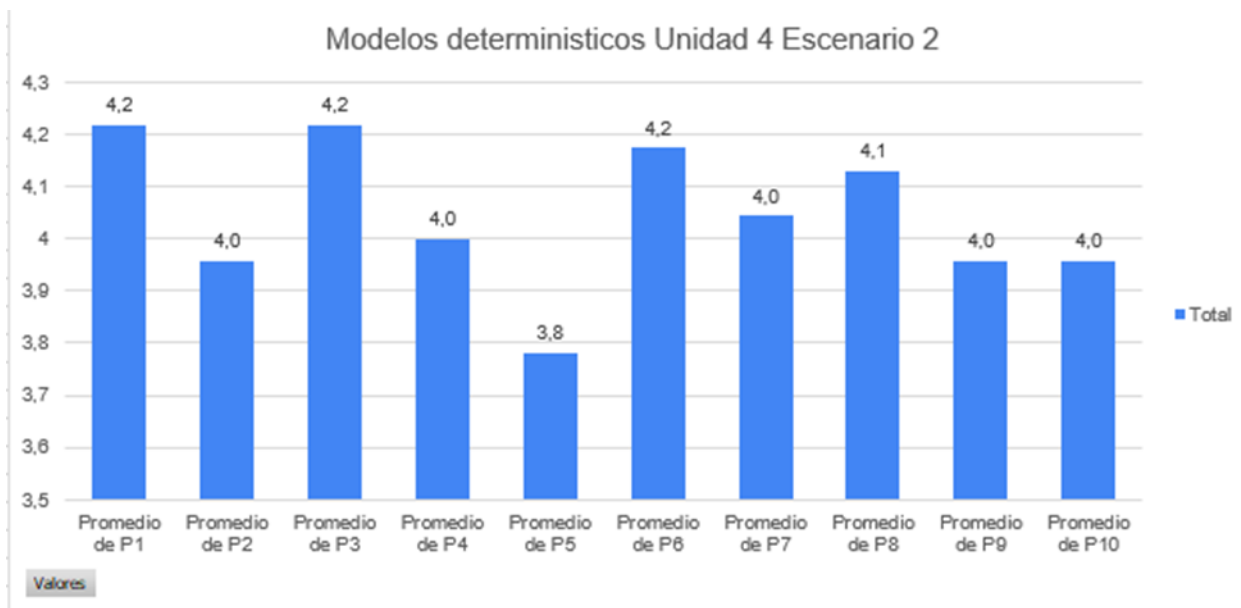
A continuación, se presentan los resultados de la evaluación heurística realizada por usuarios calificados, cada figura presenta los resultados de usabilidad de los diferentes escenarios realizados en el software y su correspondiente evaluación. La técnica utilizada para recopilar estos datos está basada en la heurística de Nielsen, el cual presenta diez principios representados en el eje de las x con la nomenclatura Promedio P1... Promedio P10. La escala definida en el eje de las y fue un valor de 0 a 5, donde 0 representa que no se cumple con el principio de usabilidad y 5 que se cumple plenamente. Por ultimo las gráficas muestra es el promedio de un grupo de 25 usuarios expertos que evaluaron la aplicación.

Figura 10. Resultados de evaluación heurística del escenario 1



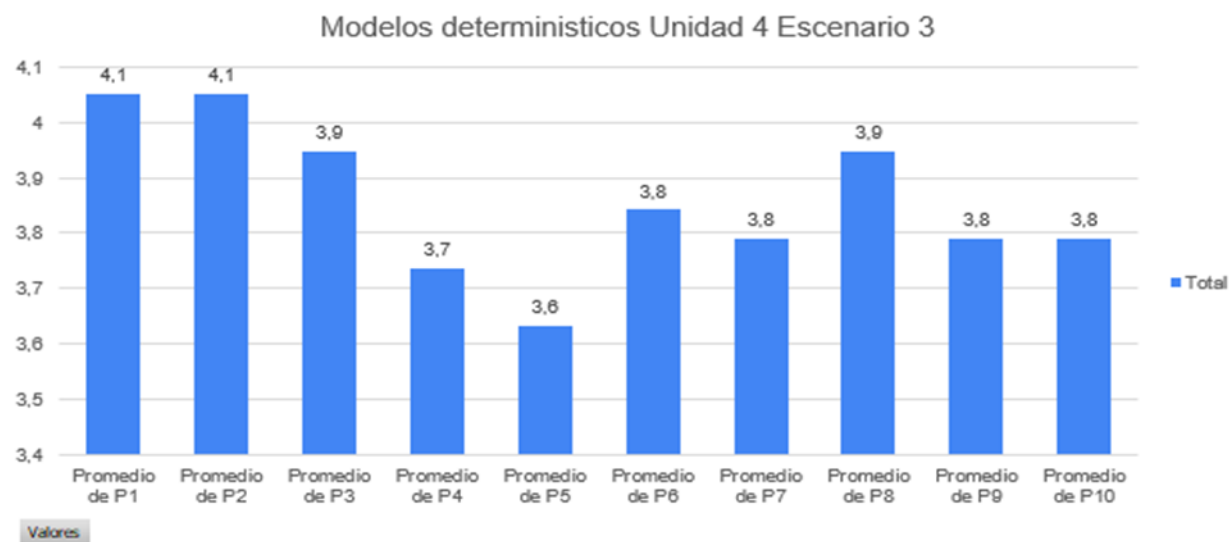
Nota. En la figura se muestran los resultados de la evaluación del escenario 1. Fuente: elaboración propia.

Figura 11. Resultados de evaluación heurística del escenario 2.



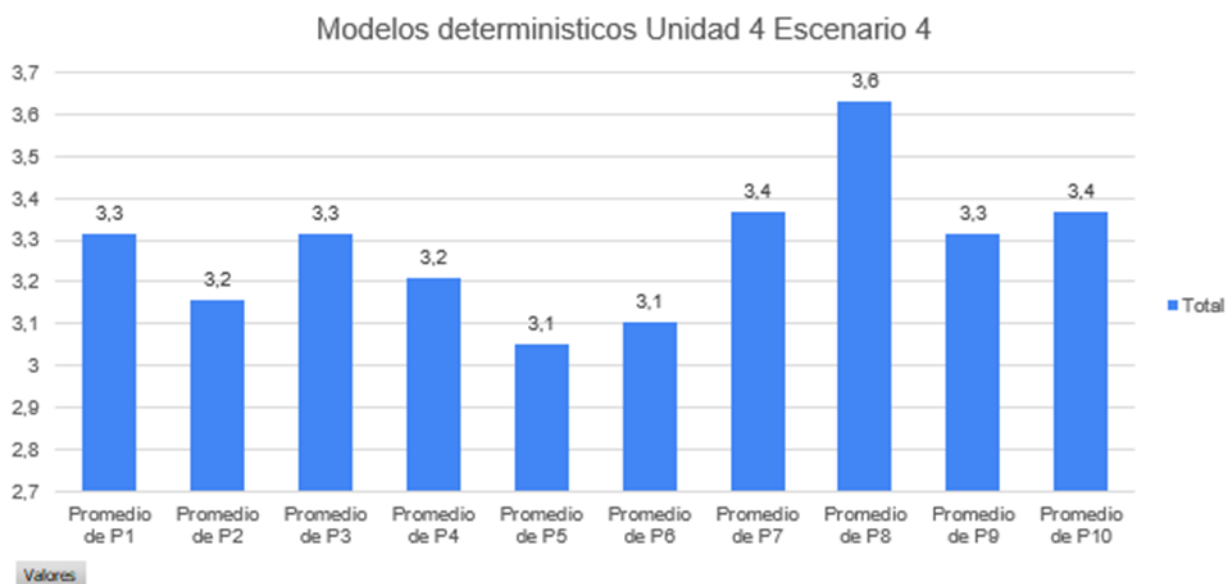
Nota. En la figura se muestran los resultados de la evaluación del escenario 2. Fuente: elaboración propia.

Figura 12. Resultados de evaluación heurística del escenario 3.



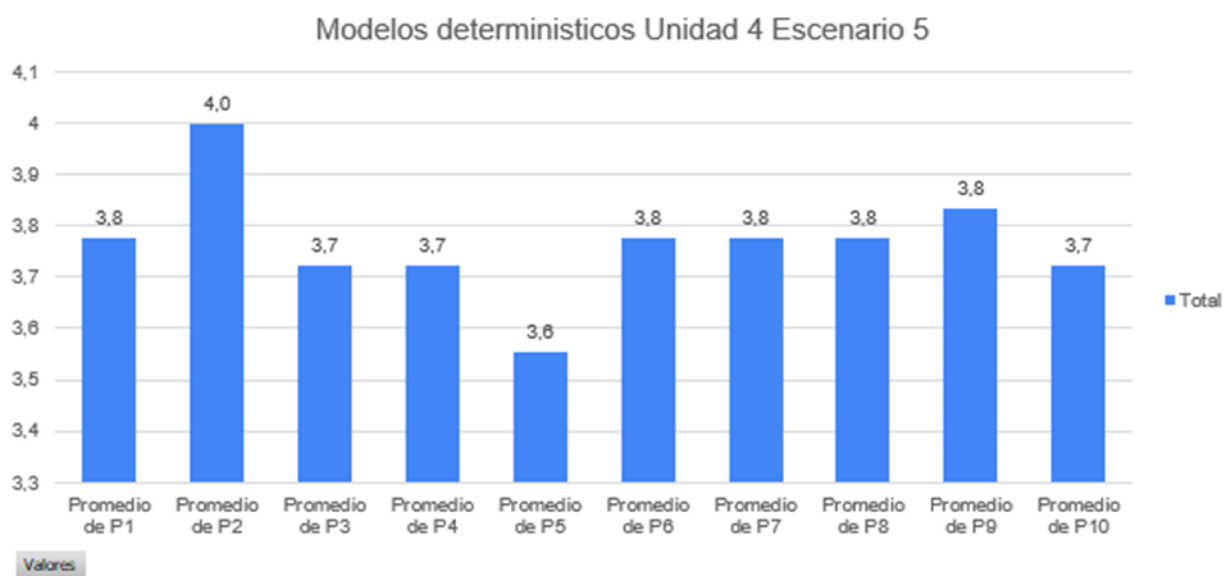
Nota. En la figura se muestran los resultados de la evaluación del escenario 2. Fuente: elaboración propia.

Figura 13. Resultados de evaluación heurística del escenario 4



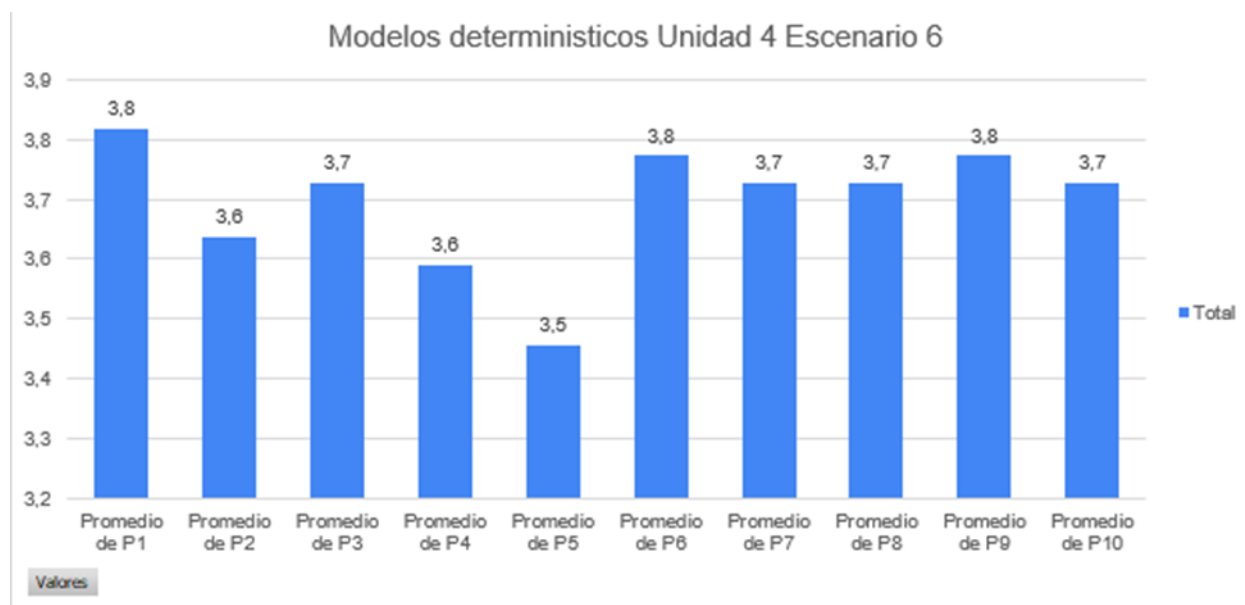
Nota. En la figura se muestran los resultados de la evaluación del escenario 4. Fuente: elaboración propia.

Figura 14. Resultados de evaluación heurística del escenario 5.



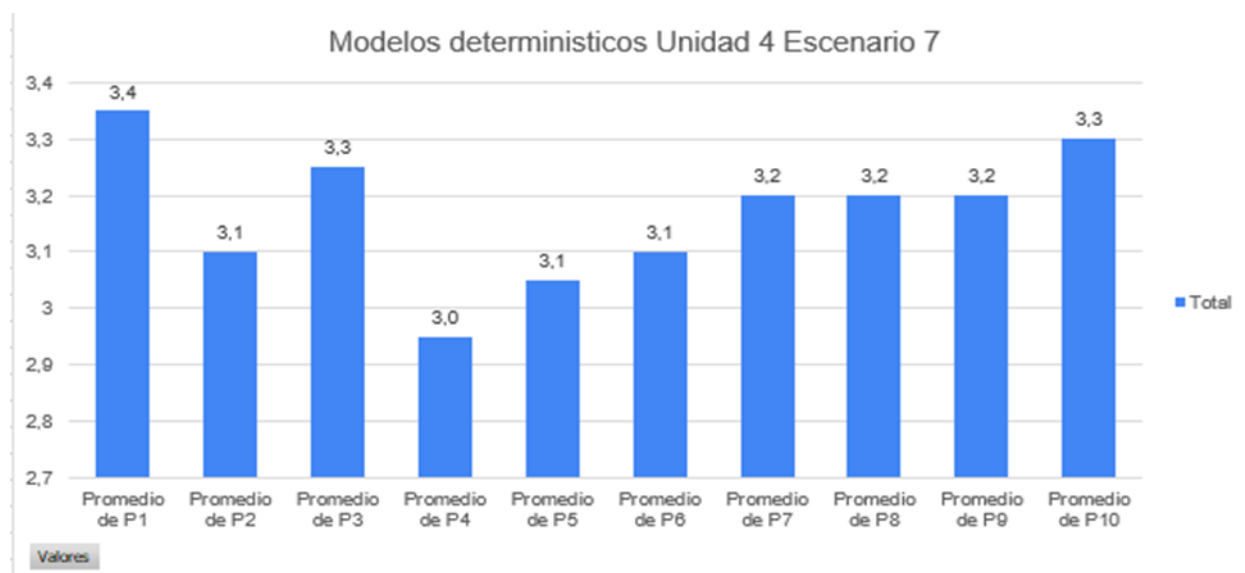
Nota. En la figura se muestran los resultados de la evaluación del escenario 5. Fuente: elaboración propia.

Figura 15. Resultados de evaluación heurística del escenario 6.



Nota. En la figura se muestran los resultados de la evaluación del escenario 6. Fuente: elaboración propia.

Figura 16. Resultados de evaluación heurística del escenario 7.



Nota. En la figura se muestran los resultados de la evaluación del escenario 7. Fuente: elaboración propia.

Los resultados de la evaluación se pueden evidenciar una deficiencia en el control de errores en la todos los escenarios, es de los puntos más críticos en conjunto de todos, esto significa los usuarios identifican dificultades y no encuentran manera de resolverlos de rápida o fácil, falta mejorar ese aspecto para que anticipe posibles causas de error, por ende, mejorar esta usabilidad para futuras actualizaciones, por general los resultados arrojan diseño y estética de mejor manera más positiva, acercándose a la enseñanza y falta mejorar en aspectos de prevención de errores para elevar la experiencia de usuario, esto ayuda a la percepción general del sistema recibida de la mejor manera. Según Pressman (2010) la prevención de errores en el desarrollo de cualquier software es esencial garantizar la fiabilidad a los usuarios y la calidad de los resultados buscados.

Conclusiones

El desarrollo del entorno virtual “UdlaVerso” para la enseñanza de programación dinámica en modelos determinísticos ha demostrado ser una solución efectiva y adaptada a las necesidades educativas actuales en el ámbito de la ingeniería. Atraves de este entorno inmersivo se logra percibir la mejora de aprendizaje complejos pudiendo los usuarios interesados utilizando explorar e interactuar activamente al aprendizaje teórico practico (Bustos A, 2010). Se utilizaron tecnologías avanzadas como el visor FireStorm y los scripts en el lenguaje de LSL (Linden Scripting Language), lo permitió crear escenarios interactivos y dinámicos para acercan a los estudiantes y personas interesada, a una experiencia de aprendizaje inmersiva. Estos escenarios enriquecen la teoría y su aplicación práctica ampliando el conocimiento, ayudando a los estudiantes experimentar, editar, y mejorar directamente con los conceptos, lo cual incrementa significativamente su comprensión y retención de conocimientos, permitiendo desarrollar una comprensión más profunda de los algoritmos y comportamientos internos de la programación dinámica (Guzmán, 2022).

Los resultados obtenidos muestran que el “Udlaverso” es una herramienta valiosa que no solo para mejorar la comprensión de las temáticas a usuarios interesados, sino que el componente innovador y alterno al aprendizaje tradicional capta la atención a querer aprender estos conceptos (Devolder, 2012). Es importante resaltar que esta

metodología puede ser replicada en otros campos de acción contrayéndolos y adaptarlos en otras áreas del conocimiento, lo que hay un potencial para transformar la enseñanza tradicional a una experiencia diferente el cual puede ser complementaria y acompañante en los procesos de enseñanza modernos (Fariñas, 2004).

Para terminar, el “Udlaverso” ofrece una respuesta oportuna a la falta de una herramienta complementaria y a la medida para el aprendizaje de conceptos complejos, sino que también amplía la visión del aprendizaje moderno, lo cual es importante también al fomento del desarrollo de nuevos campos de acciones al futuro y a la contribución de las habilidades críticas a los estudiantes. Este proyecto se establece como un referente en la aplicación de tecnología avanzada en educación, subrayando la importancia de metodologías innovadoras y escenarios de aprendizaje interactivo para abordar conceptos complejos en ingeniería y ciencias aplicadas.

Referencias Bibliográficas

- Acosta, A. S. (2019). Programación dinámica en el cálculo de la ruta óptima de una topología de red: Caso de estudio. *Revista Espacios*.
- Andrade, H. G. (2005). Teaching with rubrics: The good, the bad, and the ugly.
- Burgués, J. E. (2018). *Aprende a Modelar Aplicaciones con UML-Tercera Edición*. IT campus academy.
- Bustos A, C. C. (2010). Los entornos virtuales como espacios de enseñanza y aprendizaje.
- Ceballos, D. E. (2021). Método de selección de técnicas de levantamiento de requerimientos para el desarrollo de software con un enfoque de experiencia de usuario.
- Devolder, A. B. (2012). Los ambientes virtuales de aprendizaje: una revisión de la literatura. *Revista de Educación a Distancia*.
- Gama E, H. R. (SF). *Design Patterns Elements of Reusable Object-Oriented Software*.
- García-Peñalvo, F. J.-G.-H.-I. (2021). *Unified Modeling Language*.
- Grau, X. F. (2008). *Desarrollo orientado a objetos con UML*.
- Guzmán, M. C. (2022). La didáctica en los entornos virtuales de aprendizaje. *Revista Metropolitana de Ciencias Aplicadas*.
- López, D. A. (2018). *Programación orientada a objetos I*. Servicio de Publicaciones de la Universidad Católica de Ávila.
- Peña, V. R. (2020). *La inteligencia artificial en la educación. Dominio de las Ciencias*. Dominio de las Ciencias.
- Salamanca Aguado, V. (2020). Desarrollo de un portal educativo Web en Joomla! aplicando métrica V3. 89.
- Trujillo, D. B. (2019). Uso de simuladores 3D como estrategia tecnopedagógica para la transferencia de conocimiento en el aprendizaje de la anatomía animal.
- Y, B. (2015). La simulación expresiva como herramienta didáctica en entornos virtuales para la enseñanza de la segunda ley de Newton en grado décimo: Estudio de caso en la Institución Educativa Sor Juana Inés De La Cruz del municipio de Medellín.