



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

ESCOM

TRABAJO TERMINAL

**“Prototipo de Simulador de Laboratorio de Química
Inorgánica en Realidad Virtual”**

2024-B059

PRESENTA:

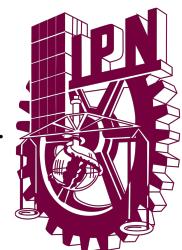
García Aguayo Marcos Martí Sandino Mictlantecuhtli

DIRECTORES:

M. en E. Saul De La O Torres Dr. Gabriel Sepúlveda Cervantes



Diciembre 2024



**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
SUBDIRECCIÓN ACADÉMICA**



No. De TT: 2024-B059

19 de diciembre de 2024

DOCUMENTO TÉCNICO

**“Prototipo de Simulador de Laboratorio de Química Inorgánica
en Realidad Virtual”**

PRESENTA:

García Aguayo Marcos Martí Sandino Mictlantecuhtli¹

DIRECTORES:

M. en E. Saul De La O Torres

Dr. Gabriel Sepúlveda Cervantes

RESUMEN:

Este trabajo presenta el desarrollo de un prototipo de simulador de laboratorio de química inorgánica en realidad virtual, orientado a la enseñanza inmersiva mediante tecnologías avanzadas como seguimiento de manos, sistemas FSM y una tabla periódica interactiva. El sistema permite realizar experimentos estructurados que abarcan balanceo de ecuaciones, creación de compuestos y simulación de reacciones químicas, integrando efectos visuales y auditivos en tiempo real. Validado por expertos.

Palabras clave: Química Inorgánica, Realidad Virtual, Simulador, Software Interactivo.

¹marcos.ab21@hotmail.com

CARTA RESPONSIVA

Que otorga visto bueno y avala la conclusión de documentación del Trabajo Terminal bajo los lineamientos establecidos por la Comisión Académica de Trabajos Terminales (CATT)

CDMX, a 19 de diciembre de 2024

**M. EN C. ANDRÉS ORTIGOZA CAMPOS
PRESIDENTE DE LA COMISIÓN ACADÉMICA DE TRABAJOS TERMINALES
P R E S E N T E**

EN ATENCIÓN A:

**M. EN A. N. MARÍA MAGDALENA SALDÍVAR ALMOREJO
SECRETARIA EJECUTIVA**

Por medio de la presente, se informa que el Trabajo Terminal Núm. 2024-B059

Que lleva por Título: Prototipo de Simulador de Laboratorio de Química

Inorgánica en Realidad Virtual

Fue concluido satisfactoriamente por:

García Aguayo Marcos Martí Sandino Mictlantecuhtli

Se avala que la documentación entregada mediante discos en formato DVD fue **revisada de manera precisa y exhaustiva** con el propósito de asegurar que los avances desarrollados bajo la supervisión de quien o quienes suscriben, hayan cumplido con lo planteado en el protocolo original, así como en lo establecido por el Documento Rector de Operación y Evaluación para los Trabajos Terminales de la ESCOM.

**ATENTAMENTE
"LA TÉCNICA AL SERVICIO DE LA PATRIA"**

M. en E. Saul De La O Torres
Director

Dr. Gabriel Sepúlveda Cervantes
Director

Advertencia

“Este documento contiene información desarrollada por la Escuela Superior de Cómputo del Instituto Politécnico Nacional, a partir de datos y documentos con derecho de propiedad y por lo tanto, su uso quedará restringido a las aplicaciones que explícitamente se convengan.”

La aplicación no convenida exime a la escuela su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen.

Información adicional sobre este reporte técnico podrá obtenerse en:

La Subdirección Académica de la Escuela Superior de Cómputo del Instituto Politécnico Nacional, situada en Av. Juan de Dios Bátiz s/n Teléfono: 57296000, extensión 52000.

Agradecimientos

Este proyecto no habría sido posible sin el apoyo, guía y motivación de muchas personas e instituciones a lo largo de este camino. A todas ellas, quiero expresar mi más profundo agradecimiento.

En primer lugar, mi gratitud infinita a mis directores, el **Maestro Saúl de la O Torres** y el **Doctor Gabriel Sepúlveda Cervantes**, por su invaluable guía, dedicación y apoyo constante. Su experiencia y compromiso no solo orientaron este trabajo, sino que también dejaron una huella imborrable en mi formación profesional.

Al **Instituto Politécnico Nacional**, por abrirme sus puertas y brindarme durante los últimos seis años un espacio para crecer, aprender y soñar. Gracias por proporcionarme las herramientas, conocimientos y valores necesarios para poner la técnica al servicio de la patria y contribuir a un mundo mejor.

A mi familia, que ha estado conmigo en las buenas, en las malas y en las peores, mi más sincero agradecimiento. A mi madre, **María Luisa Aguayo Torres**, mi mayor inspiración y guía. Su amor incondicional y esfuerzo incansable han sido la base de cada logro que he alcanzado. A mi padre, **Luis Carlos García Alaniz**, por su apoyo constante y por estar presente cuando más lo he necesitado. A mis hermanos, **Carlos Pavel Cuauhtémoc Tupac Amaru** y **María Sachenka América Libertad**, por ser una fuente constante de motivación, cariño y respaldo en cada etapa de este camino.

A mis compañeros del **Laboratorio de Realidad Virtual del CIDETEC**, por hacer cada tarde más amena con sus consejos, historias y camaradería. Gracias, **Gus, Ángel, Germán, Pepe, Lalo, Mike, Gabo, Steven, Wonka y Fer**, por convertir cada jornada de trabajo en una experiencia inolvidable, llena de aprendizajes y risas.

A la **Escuela Secundaria Diurna No. 4 "Moisés Sáenz"**, por abrirme sus puertas y permitirme llevar este proyecto a la práctica, facilitando un espacio para su desarrollo y pruebas.

Al **profesor Ernesto Ángel Uribe Pérez**, por su generosidad al compartir su vasto conocimiento en química, sus valiosas ideas y su apoyo constante. Su experiencia fue un aporte esencial para el éxito de este proyecto.

A la alumna **Castro Gordillo Miranda** del grupo 3º A, por su dedicación y tiempo en la realización de las pruebas que contribuyeron significativamente al desarrollo del simulador.

Finalmente, a todos los profesores que he tenido el honor de tener a lo largo de mi formación académica. Su dedicación y pasión por la enseñanza dejaron en mí una semilla de curiosidad y amor por el conocimiento que sigue creciendo cada día.

Este logro es el resultado de un esfuerzo colectivo y una suma de voluntades.



CARTA DE AUTORIZACIÓN DE DIFUSIÓN EN EL REPOSITORIO INSTITUCIONAL DE TESIS

100 Aniversario del CECyT 1 "Gonzalo Vázquez Vela"
 60 Aniversario del Centro Cultural "Jaime Torres Bodet"
 50 Aniversario de la ESIME Unidad Culhuacán,
 ESIA Unidad Tecamachalco y de la Escuela Superior de Turismo
 40 Aniversario del CIEMAD, CEPROBI y del CITEDI

I. IDENTIFICACIÓN DEL AUTOR E INVESTIGACIÓN

Nombre del autor (es)	GARCÍA AGUAYO MARCOS MARTÍ SANDINO MICTLANECUHTLI
Apellidos, Nombre(s)	
Correo electrónico	MARCOS.AB21@HOTMAIL.COM

Unidad Académica	
Escuela Superior De Cómputo.	
Nivel Educativo	
Nivel Superior	
Programa Académico	
Ingeniería En Sistemas Computacionales Plan 2009	
Título de la investigación	
PROTOTIPO DE SIMULADOR DE LABORATORIO DE QUÍMICA INORGÁNICA EN REALIDAD VIRTUAL	
Modalidad de la investigación	
Tesis	()
Seminario de titulación	()
<i>Tesina o trabajo final del seminario cursado</i>	()
Memoria de experiencia profesional	()
<i>Informe escrito de las actividades profesionales realizadas</i>	()
Proyecto de investigación	()
<i>Informe técnico de la propuesta o desarrollo de un nuevo material, equipo, prototipo, proceso o sistema</i>	()
Curricular	(✓)
<i>Informe de cursos o actividades correspondientes a esta modalidad de titulación</i>	(✓)
Práctica profesional	()
<i>Informe escrito avalado por la empresa u organismo donde se desarrolló la práctica</i>	()
Programa especial	()
<i>Informe de asignaturas tecnológicas propias de la rama cursada previamente</i>	()
Otro	()
<i>Resumen o versión alterna autorizada por el autor</i>	()

II. ESPECIFICACIONES DEL DOCUMENTO (FORMATO DIGITAL)

Formato del archivo "versión pública de la investigación"	PDF (✓)
Número de páginas <i>Incluyendo anexos. Excluyendo portada.</i>	128 páginas
Formato del archivo "documentos de soporte"	PDF (✓)
Número de páginas <i>Incluye: autorización de difusión y formatos institucionales</i>	126 páginas

Página 1 de 2

Av. Instituto Politécnico Nacional s/n esquina Av. Wilfrido Massieu, Unidad Profesional "Adolfo López Mateos"
 Col. San Pedro Zacatenco, Alcaldía Gustavo A. Madero, C.P. 07738, CDMX
 Tel. (55) 5729 6000 exts. 54384 y 54306 ipn.mx/bibliotecas-publicaciones/





EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

SECRETARÍA DE
SERVICIOS EDUCATIVOS
DIRECCIÓN DE BIBLIOTECAS
Y PUBLICACIONES

CARTA DE AUTORIZACIÓN DE DIFUSIÓN EN EL REPOSITORIO INSTITUCIONAL DE TESIS

100 Aniversario del CECyT 1 "Gonzalo Vázquez Vela"
60 Aniversario del Centro Cultural "Jaime Torres Bodet"
50 Aniversario de la ESIME Unidad Chihuacán,
ESIA Unidad Tecamachalco y de la Escuela Superior de Turismo
40 Aniversario del CIEMAD, CEPROBI y del CITEDI

III. AUTORIZACIÓN DE DIFUSIÓN DE LA INVESTIGACIÓN

El (los) que suscribe(n) GARCÍA AGUAYO MARCOS MARTÍ SANDINO MICTLANTECUHTLI, con fundamento en los artículos 21 y 27 de la Ley Federal del Derecho de Autor, así como lo dispuesto en los artículos 5, 7 fracción IV, 9 y 11, fracción XXI de la Ley General en materia de Humanidades, Ciencias, Tecnologías e Innovación, como titular(es) de los derechos moral y patrimonial de la investigación titulada: PROTOTIPO DE SIMULADOR DE LABORATORIO DE QUÍMICA INORGÁNICA EN REALIDAD VIRTUAL, consiento (consentimos) al Instituto Politécnico Nacional (IPN), a través de la ESCOM, así como a la Dirección de Bibliotecas y Publicaciones (DBP), para que publiquen por tiempo indefinido de forma gratuita y no exclusiva, en el Repositorio Institucional de Tesis, los contenidos de mi investigación, autorizándolo a:

- Almacenar la obra en el citado repositorio para efectos de seguridad y preservación;
- Publicar la obra con fines académicos y de investigación;
- Transformar o convertir la obra a distintos formatos o soportes electrónicos, para su accesibilidad, preservación y seguridad.

Al tenor de lo previamente expuesto, manifiesto que la obra es original y no infringe los derechos de la propiedad intelectual de otras personas o entidades, por lo que excluyo al Instituto Politécnico Nacional de todo tipo de responsabilidad, ya sea civil, administrativa o penal en virtud de su contenido.

La obra se pondrá a disposición en el Repositorio Institucional de Tesis (<https://tesis.ipn.mx>), respetando los derechos de autor considerando como requisito la licencia de uso Creative Commons BY-NC-ND 4.0 © ⓘ ⓘ ⓘ.

El autor entiende que, de no autorizar la publicación de su tesis, ésta será catalogada de manera interna en la biblioteca de la ESCOM únicamente a nivel referencial para efectos de control. El autor podrá solicitar por escrito el retiro de su obra del Repositorio Institucional de Tesis, ante su Unidad Académica si lo considera conveniente; por su parte, el Instituto Politécnico Nacional podrá en cualquier momento retirar la publicación si existieran reclamaciones de terceros que afirmen ser titulares.

Así mismo, manifiesto bajo protesta de decir verdad que he leído y conozco el Aviso de Privacidad disponible en la página electrónica de la Dirección de Bibliotecas y Publicaciones, <https://www.ipn.mx/bibliotecas-publicaciones/>

Ciudad de México, a 19 de Diciembre de 2024

GARCÍA AGUAYO MARCOS MARTÍ SANDINO MICTLANTECUHTLI

Nombre(s) completo y firma del autor(es)

Página 2 de 2



2024
Felipe Carrillo
PUERTO

Índice general

Resumen	II
Carta Responsiva	III
Advertencia	IV
Agradecimientos	V
Carta De Autorización De Difusión En El Repositorio Institucional De Tesis	VI
1. Introducción	1
1.1. Antecedentes	1
1.2. Planteamiento del Problema	2
1.3. Solución Propuesta	2
1.4. Objetivos	3
1.4.1. General	3
1.4.2. Específicos	3
1.5. Justificación	3
1.6. Estado del arte	4
1.6.1. VRLab Academy	4
1.6.2. Garg Lab VR Chem	5
1.6.3. PraxiLabs	6
1.6.4. Laboratorio Virtual de Experimentación Química UPM	7
1.6.5. EVA Tech	8
1.6.6. HoloLab Champions	9
1.6.7. Tabla Comparativa	10
2. Marco Teórico	11
2.1. Química	11
2.1.1. Química Inorgánica	11
2.1.2. Reacciones Químicas	11
2.1.3. Estequiometría	12
2.1.4. Cinética Química	13
2.1.5. Ley de Rapidez	13
2.1.6. Leyes ponderales	13
2.2. Simulador	14
2.2.1. Elementos de una simulación [10]	14
2.2.2. Beneficios	15
2.2.3. Limitaciones	15

2.3.	Realidad Virtual (RV)	16
2.3.1.	Componentes de un Sistema de RV	16
2.3.2.	Inmersión e Interacción	16
2.3.3.	Aplicaciones de la RV en la Educación	17
2.4.	Laboratorio Virtual	17
2.4.1.	Importancia de los Laboratorios Virtuales	17
2.4.2.	Limitaciones en el uso de laboratorios virtuales	18
2.5.	Modelado y Diseño	19
2.5.1.	Modelado 3D	19
2.5.2.	Modelado 2D	20
3.	Análisis General	21
3.1.	Metodología	21
3.2.	Análisis de Requerimientos	23
3.2.1.	Requerimientos Funcionales	23
3.2.2.	Requerimientos No Funcionales	24
3.2.3.	Reglas de Negocio	25
3.2.4.	¿Como debe venderse?	26
3.2.5.	Limitaciones del Sistema	27
3.3.	Tecnologías de Desarrollo	28
3.3.1.	Modelado 3D y Animación	28
3.3.2.	Motor de Videojuegos	29
3.3.3.	Integración de Oculus Meta Quest 2 y Oculus Meta Quest Pro	32
3.3.4.	Handtracking	34
3.4.	Análisis Económico	34
3.4.1.	Ánálisis por Puntos de Función	34
3.4.2.	Análisis de costos	39
3.5.	Modelado del Sistema	42
3.5.1.	Casos de Uso	42
3.5.2.	Diagrama de Clases	59
4.	Desarrollo	60
4.1.	Estructura del Simulador	60
4.2.	Ambiente Virtual	62
4.2.1.	Diseño del entorno 3D	62
4.2.2.	Efectos Visuales (VFX)	64
4.2.3.	Efectos de Sonido	65
4.2.4.	Instrucciones guiadas con voz	65
4.3.	Diseño de Interfaz de Usuario	65
4.3.1.	Hub de selección	65
4.3.2.	Información principal	66
4.3.3.	Elementos	67
4.3.4.	Compuestos	67
4.3.5.	Área de creación	67
4.3.6.	Teclado numérico interactivo	68
4.3.7.	Pantalla final	68
4.4.	Interacción y Gestos	69
4.4.1.	Interacción con el sistema	69
4.4.2.	Gestos específicos	70
4.5.	Integración del SDK de Oculus	71

5. Implementación	72
5.1. Fases de los Experimentos	72
5.1.1. Introducción	72
5.1.2. Balanceo de Ecuaciones	72
5.1.3. Selección y Creación	73
5.1.4. Experimentación	73
5.1.5. Explicación	73
5.2. Escenas Principales	74
5.2.1. Hub	74
5.2.2. Tutorial	74
5.2.3. Experimento 1	75
5.2.4. Experimento 2	76
5.2.5. Experimento 3	76
5.2.6. Experimento 4	77
5.3. Acciones del Usuario	78
5.3.1. Manipulación de Objetos	78
5.3.2. Selección de Elementos en la Tabla Periódica	79
5.3.3. Creación y Manipulación de Compuestos	79
5.3.4. Encendido de Mecheros y Materiales	80
5.3.5. Eliminación de Elementos	80
5.3.6. Acciones Relacionadas con Líquidos	81
6. Presentaciones y Pruebas del Prototipo	82
6.1. Presentaciones Institucionales	82
6.2. Invitaciones Externas	83
6.3. Colaboración y Validación Académica	85
7. Conclusiones	86
8. Trabajo a Futuro	87
A. Descripción Detallada de los Experimentos	88
A.1. <i>De Básico a Ácido</i>	88
A.2. <i>La lámpara de magnesio</i>	91
A.3. <i>Nieve Química</i>	93
A.4. <i>La Bruja de Bromo</i>	95
B. Scripts Relevantes	97
B.1. MenuController.cs	97
B.2. Experiment_1_Manager.cs	100
B.3. Experiment_2_Manager.cs	102
B.4. Experiment_3_Manager.cs	105
B.5. Experiment_4_Manager.cs	107
B.6. SubstanceTracker.cs	108
B.7. Liquid_Level.cs	112
Bibliografía	116

Índice de figuras

1.1. Esquema de Solución.	2
1.2. Simulador VRLab Academy.	4
1.3. Aplicación VR Chem.	5
1.4. Praxi Labs - Laboratorio de Química Inorgánica.	6
1.5. Experimentación Química UPM.	7
1.6. Módulo Química.	8
1.7. HoloLab Champions - Chemiluminescence.	9
 3.1. Representación del modelo iterativo y modelo iterativo e incremental	21
3.2. Cronograma de Actividades	23
3.3. Modelo de un átomo en Blender	28
3.4. Game Object en Unity	30
3.5. Componentes de un Elemento en Unity	30
3.6. ScriptableObject de un Elemento en Unity	31
3.7. Oculus Meta Quest 2	32
3.8. Oculus Meta Quest Pro	32
3.9. Ejemplo de Handtracking	34
3.10. Tabla de lenguajes de puntos de función - QSM[19]	37
3.11. Laprop MSI Katana 15 B12V	40
3.12. Sistemas Compatibles	40
3.13. Diagrama de casos de uso para el sistema	42
3.14. Diagrama de Clases	59
 4.1. Assets del Simulador	60
4.2. Modelos de Elementos Químicos	62
4.3. Tabla Periódica	63
4.4. Instrumental de Laboratorio	63
4.5. Compuestos Químicos	64
4.6. Efectos de Partículas	64
4.7. IU Hub	66
4.8. IU Principal	66
4.9. IU Elementos	67
4.10. IU Compuestos	67
4.11. IU Creación	68
4.12. IU Numpad	68
4.13. IU Pantalla Final	69
4.14. Sujeción completa	69
4.15. Gesto de pinchar	70
4.16. Interacción con botones cercanos	70

4.17. Gesto para centrar el entorno	71
5.1. Flujo de la escena Hub	74
5.2. Flujo de la escena Tutorial	75
5.3. Flujo de la escena Experimento 1	75
5.4. Flujo de la escena Experimento 2	76
5.5. Flujo de la escena Experimento 3	76
5.6. Flujo de la escena Experimento 4	77
5.7. Diversas Interacciones con objetos	78
5.8. Tabla Periódica	79
5.9. Zona de Creación de compuestos	79
5.10. Encendido diverso	80
5.11. Eliminación de un elemento	80
5.12. FSM Comportamiento de Líquidos	81
5.13. Diversas Interacciones de Líquidos	81
6.1. Presentaciones Institucionales	83
6.2. Presentación Isidro Fabela	83
6.3. Presentación en Cinvesniños	84
6.4. Presentación UEVE	84
6.5. Reuniones en Secundaria 4	85

Índice de tablas

1.1.	Comparativa de simuladores de laboratorios virtuales	10
2.1.	Comparativa de los diferentes softwares para el modelado 3D	19
2.2.	Comparativa de los diferentes softwares para el modelado 2D	20
3.1.	Comparación de especificaciones entre Oculus Meta Quest 2 y Oculus Meta Quest Pro	33
3.2.	Cálculo de Puntos de Función (PF)	35
3.3.	COCOMO básico	38
3.4.	Costos de Desarrollo	39
3.5.	Costos de Infraestructura	41
3.6.	Costos de Licencias	41
3.7.	Descripción del caso de uso Seleccionar Experimento	44
3.8.	Descripción del caso de uso Iniciar Tutorial	45
3.9.	Descripción del caso de uso Balancear Ecuación Química	46
3.10.	Descripción del caso de uso Seleccionar Elemento Químico	48
3.11.	Descripción del caso de uso Crear Compuesto	49
3.12.	Descripción del caso de uso Calentar Compuesto	51
3.13.	Descripción del caso de uso Enfriar Compuesto	52
3.14.	Descripción del caso de uso Combinar Sustancias	53
3.15.	Descripción del caso de uso Desechar elemento	54
3.16.	Descripción del caso de uso Experimentación y Reacción	56
3.17.	Descripción del caso de uso Main Menu	57
3.18.	Descripción del caso de uso Colocar instrumental en soporte	58

CAPÍTULO 1

Introducción

1.1. Antecedentes

En las últimas décadas, el avance tecnológico ha permeado de manera significativa en diversos ámbitos de la sociedad, transformando la forma en que interactuamos con el mundo que nos rodea. Uno de los campos donde esta revolución ha sido más evidente es en el ámbito educativo, donde la integración de la tecnología se ha convertido en un tema de relevancia ineludible. Brito [1] resalta que la emergencia de la cultura digital ha traído consigo profundas transformaciones en la forma en que concebimos la producción, circulación y apropiación del conocimiento, desafiando las estructuras y valores culturales previamente establecidos.

A lo largo del siglo XX, se han iniciado programas para incorporar herramientas tecnológicas en las escuelas de educación básica en México, pero su éxito tanto en términos operativos como pedagógicos ha sido limitado. Esto se atribuye, en parte, a la rigidez de las instituciones escolares, las cuales han enfrentado dificultades para replantear los modelos de socialización que tradicionalmente han prevalecido en el ámbito educativo[2].

La llegada de internet, las computadoras y los dispositivos digitales ha abierto un abanico de posibilidades en la educación. La revolución tecnológica de las últimas cuatro décadas ha llevado a la necesidad de integrar dispositivos de comunicación e información en los procesos educativos, considerándose incluso como la única vía para expandir el acceso al conocimiento de manera eficiente y veloz[2].

En este entorno, la educación STEM (Ciencia, Tecnología, Ingeniería y Matemáticas), se presenta como un pilar fundamental. Este enfoque interdisciplinario no solo se limita al ámbito académico, sino que también incorpora contextos y situaciones relevantes para la vida cotidiana de los estudiantes, así como los materiales necesarios para su comprensión y aplicación[3]. Desde la perspectiva de Zamorano Escalona, García Cartagena y Reyes González [4], la educación STEM, o STEAM al incluir las Artes, busca nutrir a la industria científica y tecnológica con recursos humanos creativos. Esto se logra al aumentar el interés de los estudiantes y al desarrollar habilidades esenciales del siglo XXI, fundamentales para estimular el progreso científico-tecnológico, a través de una integración interdisciplinaria de ciencias, tecnología, matemáticas, artes e ingeniería que vincula los contenidos con las experiencias de vida de los estudiantes.

La introducción de la realidad virtual (RV) en este contexto educativo no solo representa un avance tecnológico, sino una oportunidad para superar barreras y maximizar el potencial de la enseñanza. Aunque es cierto que la accesibilidad a esta tecnología puede ser un desafío, su

implementación estratégica abre un abanico de posibilidades educativas inexploradas. La RV no solo ofrece experiencias inmersivas, sino que también brinda la capacidad de simular situaciones y entornos complejos. Esto la convierte en una herramienta invaluable para la capacitación, investigación y creación de soluciones innovadoras, no solo en el ámbito académico sino también en la industria y el mundo profesional [5].

1.2. Planteamiento del Problema

Ante la falta de interés y comprensión en la química inorgánica entre los estudiantes, se plantea el desarrollo de un prototipo de simulador de laboratorio en realidad virtual. Esta herramienta busca ofrecer una experiencia educativa más interactiva y atractiva, permitiendo a los estudiantes interactuar con elementos químicos y comprender conceptos tanto básicos como clave en un entorno virtual controlado. El objetivo es revitalizar el interés por esta materia y mejorar el aprendizaje de esta disciplina.

1.3. Solución Propuesta

Un prototipo de simulador de laboratorio de química inorgánica en realidad virtual. Los usuarios podrán realizar experimentos específicos, interactuar con los elementos y observar las reacciones en un entorno controlado.

- Prototipo de Simulador en Realidad Virtual de Química Inorgánica que consta de:
 - Tabla periódica interactiva.
 - Interfaz de usuario.
 - Algoritmos de simulación.
 - Ejercicios de química inorgánica.
- Documentación Técnica.
- Manual de Usuario.
- Pruebas y validación.



Figura 1.1: Esquema de Solución.

1.4. Objetivos

1.4.1. General

Desarrollar un prototipo de simulador en realidad virtual que permita a los usuarios interactuar con elementos químicos, combinarlos y observar sus reacciones en un entorno virtual controlado. Este simulador proporcionará una representación visual y práctica de los conceptos clave de la química inorgánica, con el propósito de complementar la comprensión teórica sin sustituir la experimentación en el laboratorio. Limitado a un contexto demostrativo, no está diseñado para su implementación en entornos educativos formales.

1.4.2. Específicos

- I Modelado e integración de una tabla periódica interactiva para una selección dinámica de elementos químicos.
- II Diseño de la Interfaz de Usuario (IU) para una interacción intuitiva con elementos químicos.
- III Desarrollo e implementación de algoritmos de simulación para la experimentación virtual y observación de reacciones químicas interactivas.
- IV Crear ejercicios de complejidad progresiva para el aprendizaje de conceptos de química inorgánica.
- V Pruebas y validación del simulador para una experiencia de usuario satisfactoria.

1.5. Justificación

En el ámbito educativo, se identifica la necesidad apremiante de enriquecer la enseñanza de las ciencias, específicamente la química, entre los jóvenes. A pesar de la disponibilidad de múltiples fuentes de información y recursos audiovisuales para complementar el aprendizaje, persiste la preocupación por la falta de interés de un considerable número de jóvenes en las ciencias. Esta apatía hacia la disciplina puede estar obstaculizando la comprensión profunda y el entusiasmo por aprender sobre la química inorgánica.

En respuesta a este desafío, el proyecto propone una solución innovadora. Mediante una experiencia educativa e interactiva en realidad virtual, se busca abordar la falta de interés al ofrecer una perspectiva diferente y cautivadora de la disciplina. Al proporcionar a los jóvenes una herramienta que les permita interactuar con elementos químicos, combinarlos y observar sus reacciones en un entorno virtual controlado, se fomenta una comprensión más práctica y atractiva de los conceptos.

Esta iniciativa no solo complementa el aprendizaje teórico, sino que también busca revitalizar la motivación intrínseca de los estudiantes hacia las ciencias, en particular la química. Al hacerlo, se aspira a cultivar un ambiente de aprendizaje más estimulante y participativo, que inspire a los estudiantes a explorar y comprender de manera más profunda los principios fundamentales de la química inorgánica.

1.6. Estado del arte

1.6.1. VRLab Academy

VRLab Academy se destaca como una empresa líder en innovación tecnológica educativa, especializada en el desarrollo de contenido de experimentos de realidad virtual con propósitos científicos y educativos. Su compromiso se centra en proporcionar a los estudiantes una experiencia de laboratorio virtual auténtica que les permita llevar a cabo experimentos y mejorar sus habilidades en un entorno interactivo, seguro y educativo.



Figura 1.2: Simulador VRLab Academy.

El enfoque de VRLab Academy se basa en la utilización de datos en tiempo real, fórmulas científicas verídicas y resultados reales. Los estudiantes tienen la capacidad de guardar y compartir los datos obtenidos durante sus experimentos.

La plataforma de VRLab Academy ofrece dos versiones de sus experimentos: VR y PC. La versión de realidad virtual requiere un dispositivo de realidad virtual como Oculus o HTC, mientras que la versión para PC solo necesita una computadora o portátil con conexión a Internet. Esta variedad de opciones permite adaptarse a las diferentes necesidades y recursos disponibles para los estudiantes.

1.6.2. Garg Lab VR Chem

Grag Lab VR Chem es una iniciativa desarrollada por el equipo de químicos orgánicos de la Universidad de California, Los Ángeles (UCLA). Esta plataforma de realidad virtual ha sido creada con el propósito de mejorar el aprendizaje de la química orgánica, ofreciendo a los estudiantes la posibilidad de interactuar con modelos moleculares en un entorno tridimensional.

La aplicación guía a los estudiantes a través de un entorno virtual donde pueden interactuar con modelos moleculares en 3D. Cada molécula viene acompañada de información relevante sobre su función y su presencia en la vida cotidiana, proporcionando contexto adicional para el aprendizaje.



Figura 1.3: Aplicación VR Chem.

VR Chem ofrece una ventaja significativa en la comprensión de las estructuras moleculares en 3D, ya que permite a los estudiantes modelar moléculas complejas de una manera interactiva. La aplicación es compatible con una variedad de dispositivos de realidad virtual, incluyendo HTC VIVE, Windows Mixed Reality, Oculus Rift, Oculus Quest, PlayStation VR y Valve Index.

1.6.3. PraxiLabs

PraxiLabs es una plataforma educativa que ofrece laboratorios virtuales interactivos en diversas disciplinas científicas, incluyendo química. Esta innovadora plataforma permite a los estudiantes llevar a cabo experimentos virtuales de laboratorio sin la necesidad de contar con equipo físico. De esta manera, PraxiLabs ofrece una experiencia práctica segura y accesible desde cualquier lugar con conexión a Internet.

Se trata de un laboratorio virtual en 3D que ofrece simulaciones de física, química y biología, diseñado para facilitar el proceso de enseñanza de la ciencia tanto para educadores como para estudiantes.

PraxiLabs funciona en computadoras de escritorio y portátiles.

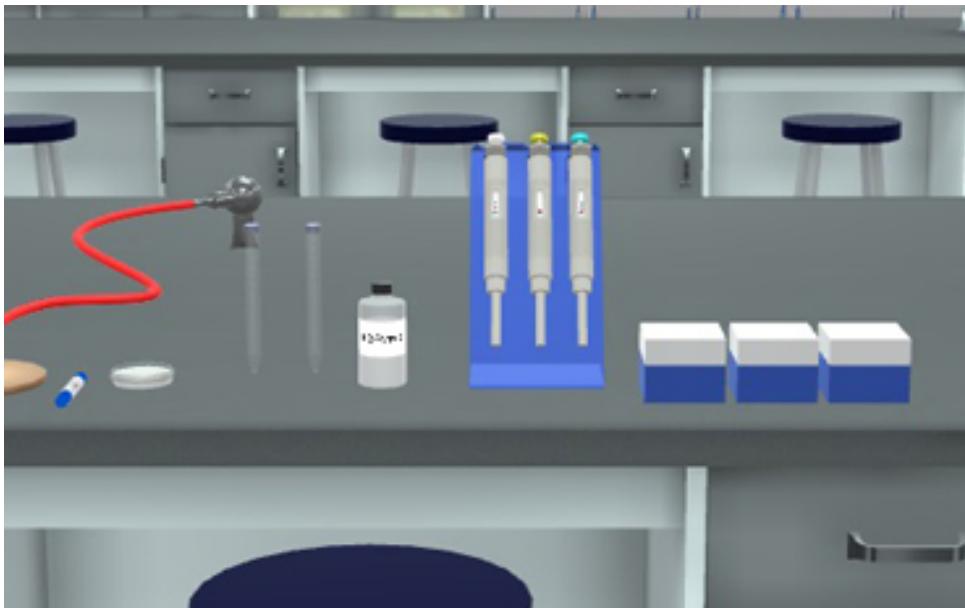


Figura 1.4: Praxi Labs - Laboratorio de Química Inorgánica.

Se requiere una conexión a Internet confiable para cargar los experimentos. La velocidad de la conexión determina la rapidez de carga. Una vez que el experimento está cargado, no es necesario seguir conectado a Internet, a menos que se utilicen archivos multimedia proporcionados.

1.6.4. Laboratorio Virtual de Experimentación Química UPM

El Laboratorio Virtual de Experimentación Química de la UPM es una herramienta educativa que simula un laboratorio real para realizar prácticas de análisis de elementos tóxicos en muestras de suelo. Esta alternativa virtual ofrece mayor seguridad y eficiencia que las prácticas presenciales, ya que reduce riesgos y tiempos de experimentación.

Los estudiantes participan en dos etapas de práctica. En la primera, acceden a material audiovisual y al quimitrivial-UPM desde mesas con ordenadores, además de utilizar un laboratorio equipado con instrumentos básicos de química. Una vez completada esta fase, acceden a los módulos de preparación e instrumentación, donde realizan un procedimiento real para determinar elementos tóxicos en muestras de suelo contaminado. Esto incluye la mineralización de la muestra en un horno de microondas y el análisis por ICP-AES.

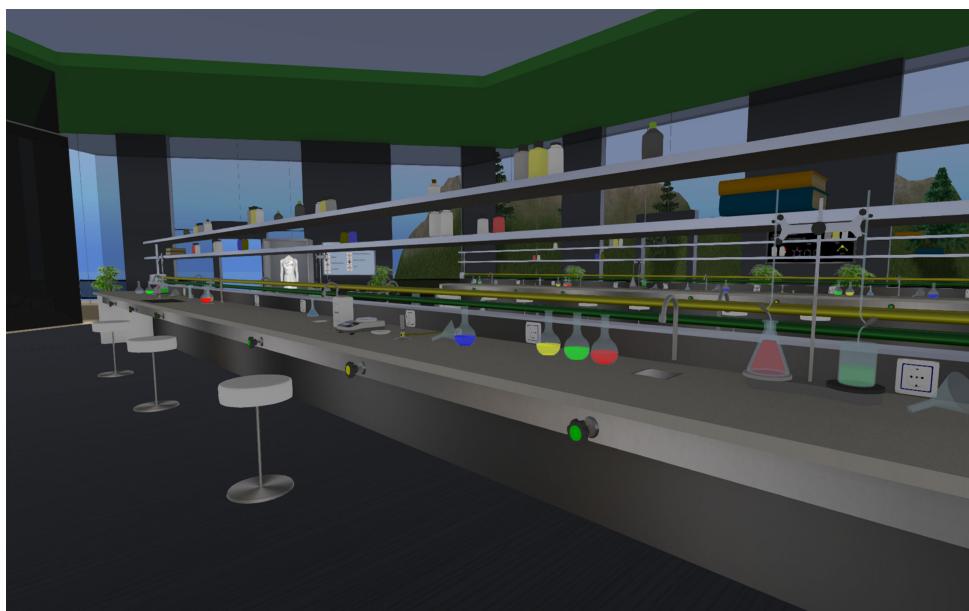


Figura 1.5: Experimentación Química UPM.

Los estudiantes ingresan al laboratorio virtual desde sus ordenadores utilizando sus cuentas institucionales UPM con el Sistema de Inicio de Sesión Único (SSO). La plataforma, construida en Unity 3D y exportada en WebGL, es accesible a través de la mayoría de los navegadores comerciales, sin descargas adicionales. Todo el contenido necesario se descarga automáticamente al conectarse, con una barra de carga indicando el progreso.

1.6.5. EVA Tech

Evatech es una plataforma de realidad virtual diseñada para abordar problemas mediante conocimientos científicos. Con un enfoque multidisciplinario, fusiona tecnología, creatividad, pensamiento crítico y habilidades de comunicación.

Esta innovadora herramienta promueve el desarrollo de competencias STEM a través de sesiones de trabajo en equipo en realidad virtual. Dirigida a estudiantes de secundaria y preparatoria, las sesiones interactivas de Evatech refuerzan conceptos científicos y fomentan el aprendizaje práctico y colaborativo.

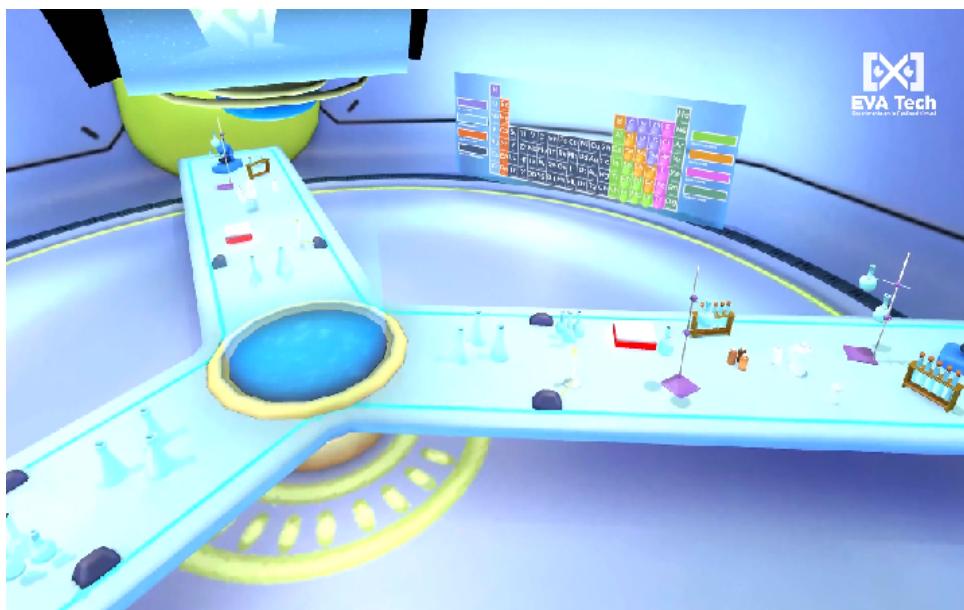


Figura 1.6: Módulo Química.

Al utilizar la realidad virtual, Evatech ofrece una experiencia educativa emocionante y efectiva, estimulando nuevas conexiones neuronales al conectar el cuerpo con el razonamiento. Su enfoque en el desarrollo de habilidades motrices y el uso de tecnologías del futuro garantiza un aprendizaje innovador y comprometido con el progreso educativo.

1.6.6. HoloLab Champions

HoloLAB Champions es un programa de juegos de realidad virtual que transporta a los jugadores a un laboratorio virtual, donde aprenden habilidades prácticas reales para realizar experimentos. Los jugadores compiten en minilabs organizados por Earl, el presentador holográfico con un sentido del humor ingenioso. Diseñado para estudiantes de 14 a 18 años, este juego de un solo jugador puede ser utilizado en entornos grupales en el aula, donde un estudiante juega mientras otros observan, brindan asistencia y toman notas, con la guía del maestro.

El juego consta de dos episodios de 30 a 40 minutos cada uno, diseñados para enseñar habilidades básicas de laboratorio, procedimientos y protocolos. El primer episodio, Chemiluminescence, enseña a los estudiantes a mezclar ingredientes líquidos y sólidos para crear una solución química brillante. El segundo episodio, Identify Unknowns, desafía a los jugadores a identificar correctamente diversas sustancias con información de referencia limitada.



Figura 1.7: HoloLab Champions - Chemiluminescence.

Financiado parcialmente por una subvención SBIR del Instituto de Ciencias de la Educación del Departamento de Educación de los Estados Unidos, HoloLAB Champions fue desarrollado en asociación con estudiantes, educadores y la RAND Corporation, siguiendo un proceso iterativo para garantizar su efectividad y utilidad educativa.

1.6.7. Tabla Comparativa

Simulador	Realidad Virtual (VR)	Hand Tracking	Etapas Integradas	Idioma	Público Objetivo
VRLab Academy	✓	✗	Visualización, Experimentación	Varios	Estudiantes de Preparatoria y Universidad
Garg Lab VR Chem	✓	✗	Visualización	Inglés	Estudiantes de pregrado
PraxiLabs	✗	✗	Experimentación	Inglés y Árabe	Estudiantes de ciencias
Laboratorio Virtual de Experimentación Química UPM	✗	✗	Balanceo de ecuaciones, Experimentación	Español	Estudiantes universitarios
EVA Tech	✓	✗	Visualización	Español	Estudiantes de Nivel Básico
HoloLab Champions	✓	✗	Experimentación	Inglés	Adolescentes y Adultos
TT-2024-B059	✓	✓	Visualización, Balanceo de ecuaciones, Creación de compuestos, Experimentación	Español	Adolescentes

Tabla 1.1: Comparativa de simuladores de laboratorios virtuales

CAPÍTULO 2

Marco Teórico

2.1. Química

La química es la ciencia que estudia la materia, su composición, estructura, propiedades y las transformaciones que experimenta. En el contexto de este proyecto, nos centraremos en los aspectos de la química inorgánica relevantes para los experimentos que se simularán en el entorno virtual.

2.1.1. Química Inorgánica

Constituye una de las ramas principales de la química, enfocada en el estudio de los elementos y compuestos que no contienen carbono, excluyendo así los compuestos orgánicos. Su ámbito de estudio abarca desde la comprensión de las propiedades básicas de los elementos individuales hasta la síntesis y caracterización de compuestos inorgánicos complejos.

2.1.2. Reacciones Químicas

Las reacciones químicas representan procesos en los cuales sustancias reactantes se transforman en otras sustancias, denominadas productos. Estas transformaciones se expresan mediante ecuaciones químicas, donde las fórmulas de los reactivos se encuentran en el primer miembro y las de los productos en el segundo. Los coeficientes en estas ecuaciones indican las proporciones molares de las sustancias participantes [6].

- **Reacciones de Oxidación-Reducción (Redox):** Una reacción de oxidación-reducción (redox) es un tipo de reacción química que implica la transferencia de electrones entre dos especies. La especie que pierde electrones se oxida, mientras que la especie que gana electrones se reduce. Estas reacciones son fundamentales en muchos procesos químicos y biológicos, y su comprensión es crucial para el estudio de la química.

En una reacción redox, el número de electrones perdidos por la especie que se oxida debe ser igual al número de electrones ganados por la especie que se reduce. Esto se debe a la ley de conservación de la masa, que establece que la materia no se crea ni se destruye en una reacción química, solo se transforma.

Para identificar una reacción redox, se pueden utilizar los números de oxidación, que indican el estado de oxidación de un átomo en un compuesto. El número de oxidación de

un átomo puede ser positivo, negativo o cero, y representa la carga hipotética que tendría el átomo si todos sus enlaces fueran iónicos.

- **Reacciones de Combustión:** Las reacciones de combustión son un tipo particular de reacción redox en las que una sustancia combustible reacciona con un oxidante (generalmente oxígeno) para producir óxido del combustible y liberar energía en forma de luz y calor.

Energía de activación y punto de ignición

Para que una reacción de combustión se inicie, se requiere un aporte inicial de energía, llamada energía de activación. Esta energía puede provenir de una llama, chispa o fuente de calor. Una vez iniciada la reacción, la energía liberada en forma de calor es suficiente para mantener la combustión hasta que se agote el combustible. El punto de ignición es la temperatura mínima a la que una sustancia combustible debe ser calentada para iniciar la combustión en presencia de oxígeno. Cada combustible tiene un punto de ignición específico.

- **Reacciones exotérmicas:** Liberan energía al entorno en forma de calor.
 - **Reacciones endotérmicas:** Absorben energía del entorno.
- **Reacciones de Síntesis:** Una reacción de síntesis es aquella en la que dos o más sustancias (reactivos) se combinan para formar un único producto. En general, estas reacciones siguen el patrón: $A + B \rightarrow AB$

2.1.3. Estequiométría

La estequiometría es una herramienta esencial en química que permite analizar las relaciones cuantitativas entre reactivos y productos en una reacción química. Se basa en el concepto de mol, que es la unidad fundamental de cantidad de sustancia en el Sistema Internacional de Unidades (SI). Un mol de cualquier sustancia contiene el mismo número de entidades elementales (átomos, moléculas, iones, etc.), conocido como número de Avogadro ($6,022 \times 10^{23}$)[7].

La estequiometría se utiliza para:

Balancear ecuaciones químicas: Ajustar los coeficientes de una ecuación química para que el número de átomos de cada elemento sea el mismo en ambos lados de la ecuación, cumpliendo así con la ley de conservación de la masa.

Calcular relaciones molares: Determinar la cantidad de moles de una sustancia que reaccionan con una cantidad dada de otra sustancia, o la cantidad de moles de producto que se forman a partir de una cantidad dada de reactivo.

Calcular relaciones de masa: Convertir entre masa y moles de una sustancia utilizando la masa molar, que es la masa de un mol de esa sustancia.

Calcular el rendimiento de una reacción: Determinar la cantidad de producto que se obtiene en una reacción química en comparación con la cantidad teórica que se debería obtener según la estequiometría.

La estequiometría es una herramienta fundamental en el laboratorio, ya que permite a los químicos diseñar experimentos, predecir los resultados y analizar los datos obtenidos.

2.1.4. Cinética Química

La cinética química es el estudio de la velocidad de las reacciones químicas y los factores que la afectan. La velocidad de una reacción se define como el cambio en la concentración de un reactivo o producto por unidad de tiempo[8].

Los factores que afectan la velocidad de una reacción química incluyen:

- **Concentración de los reactivos:** En general, la velocidad de una reacción aumenta a medida que aumenta la concentración de los reactivos. Esto se debe a que una mayor concentración de reactivos significa que hay más moléculas disponibles para colisionar y reaccionar.
- **Temperatura:** La velocidad de una reacción generalmente aumenta a medida que aumenta la temperatura. Esto se debe a que las moléculas se mueven más rápido a temperaturas más altas, lo que aumenta la frecuencia y la energía de las colisiones.
- **Catalizadores:** Un catalizador es una sustancia que aumenta la velocidad de una reacción química sin consumirse en ella. Los catalizadores funcionan al proporcionar un camino de reacción alternativo con una menor energía de activación.

2.1.5. Ley de Rapidez

La ley de rapidez, también conocida como ecuación de velocidad, describe la relación matemática entre la velocidad de una reacción química y la concentración de los reactivos. Esta ley se expresa de la siguiente manera:

$$\text{Velocidad de reacción} = k[A]^m[B]^n\dots$$

Donde:

k: Es la constante de velocidad de la reacción, que depende de la temperatura y la naturaleza de los reactivos.

[A], [B], ...: Son las concentraciones molares de los reactivos A, B, etc.

m, n, ...: Son los órdenes de reacción respecto a cada reactivo, que indican cómo la concentración de cada reactivo afecta la velocidad de la reacción. Estos órdenes de reacción se determinan experimentalmente.

La ley de rapidez es una herramienta fundamental en la cinética química, ya que permite predecir cómo cambiará la velocidad de una reacción al variar las concentraciones de los reactivos.

2.1.6. Leyes ponderales

También conocidas como leyes de las combinaciones químicas, son principios fundamentales que describen las relaciones cuantitativas entre los reactantes y productos en las reacciones químicas. Estas leyes fueron establecidas por John Dalton a principios del siglo XIX y sentaron las bases para la estequiometría, la rama de la química que se ocupa de las relaciones cuantitativas en las reacciones químicas[9].

Ley de la Conservación de la Masa

En toda reacción química ordinaria, la masa total de los reactantes es igual a la masa total de los productos. Esto significa que la materia no se crea ni se destruye durante la reacción, solo se transforma de una forma a otra.

Ley de las Proporciones Definidas

Los elementos químicos se combinan para formar compuestos en proporciones fijas y definidas. Esto significa que la composición de un compuesto puro siempre es la misma, independientemente de cómo se haya obtenido.

Ley de las Proporciones Múltiples

Cuando dos elementos forman más de un compuesto, las masas de uno de ellos que se combinan con una masa fija del otro elemento guardan una relación sencilla de números enteros.

2.2. Simulador

Los simuladores son herramientas fundamentales en campos diversos, donde ofrecen una recreación detallada y precisa de sistemas complejos. Al recrear sensaciones físicas y comportamientos específicos, los simuladores permiten a los usuarios experimentar situaciones de manera segura y controlada. Desde entrenamientos de pilotos hasta pruebas de diseño de productos, la simulación proporciona un entorno virtual que facilita la comprensión del comportamiento de sistemas complejos y la evaluación de estrategias de operación. Además, los simuladores son una parte crucial de la innovación y el desarrollo, al permitir experimentar con diferentes escenarios sin arriesgar recursos o vidas humanas.

2.2.1. Elementos de una simulación [10]

- I. Definición del Sistema: Antes de adentrarnos en la simulación del sistema, es crucial realizar un análisis preliminar exhaustivo. Este análisis nos permite entender la interacción del sistema con otros, identificar restricciones, determinar las variables involucradas y sus relaciones, así como establecer las medidas de efectividad y los resultados esperados.
- II. Formulación del Modelo: Una vez clarificados los objetivos del estudio, pasamos a la etapa de construcción del modelo. Aquí, se definen meticulosamente todas las variables que conforman el sistema, sus interrelaciones y se elaboran diagramas de flujo que retraten de manera integral el modelo en cuestión. Este proceso es fundamental para garantizar resultados precisos y útiles.
- III. Colección de Datos: La disponibilidad y accesibilidad de ciertos datos, así como la complejidad para obtener otros, pueden incidir en la formulación y desarrollo del modelo. Por ello, es crucial definir con precisión los datos necesarios para lograr los resultados deseados. Habitualmente, esta información se obtiene de registros contables, órdenes de trabajo, órdenes de compra, opiniones de expertos y, en última instancia, a través de experimentación si es imprescindible. Es fundamental esta etapa para asegurar la solidez y fiabilidad del modelo.
- IV. Implementación del Modelo en la Computadora: Una vez que el modelo está definido, el siguiente paso implica la elección entre el desarrollo en un lenguaje de programación

específico o la utilización de un paquete de software adecuado para su procesamiento en la computadora, con el fin de obtener los resultados deseados.

- V. Validación: Durante esta etapa, se pueden identificar posibles deficiencias tanto en la formulación del modelo como en los datos utilizados para alimentarlo. Las formas más comunes de validar un modelo son las siguientes:
- Evaluación por expertos de los resultados de la simulación.
 - Comparación de la precisión en la predicción de datos históricos.
 - Evaluación de la precisión en la predicción de eventos futuros.
 - Comprobación de la falla del modelo de simulación al utilizar datos que causen fallas en el sistema real.
 - Aceptación y confianza en el modelo por parte de las personas que utilizarán los resultados generados por el experimento de simulación.
- VI. Experimentación: Una vez validado el modelo, se procede a la experimentación. Esta fase implica la generación de los datos necesarios y la realización de análisis de sensibilidad de los indicadores pertinentes.
- VII. Interpretación: Una vez completada la experimentación y obtenidos los resultados de la simulación, se procede a su interpretación. En esta fase, se analizan detalladamente los datos generados y se extraen conclusiones relevantes. Basándose en esta interpretación, se toma una decisión informada.
- VIII. Documentación: Para maximizar el uso del modelo de simulación, se requieren dos tipos de documentación: técnica y manual de usuario. La documentación técnica detalla el funcionamiento interno del modelo, mientras que el manual de usuario proporciona instrucciones prácticas para su uso efectivo. Juntos, estos documentos garantizan que el modelo sea accesible y útil para todos los usuarios, facilitando su aplicación en diversas situaciones.

2.2.2. Beneficios

- Mayor seguridad: Permiten a los usuarios practicar en entornos seguros, evitando riesgos y daños.
- Menor costo: Su desarrollo y uso puede ser más económico que realizar experimentos en el mundo real.
- Mayor control: Permiten controlar las condiciones de la simulación, lo que facilita la investigación y el análisis.
- Mayor accesibilidad: Pueden ser utilizados por personas que no podrían acceder a experiencias en el mundo real.

2.2.3. Limitaciones

- Complejidad: La creación de simuladores realistas y precisos puede ser un proceso complejo y costoso.

- Fidelidad: Siempre tendrán un cierto grado de abstracción, lo que puede limitar su precisión.
- Sesgos: Pueden reflejar los sesgos de sus creadores, lo que puede llevar a resultados inexactos o discriminatorios.

2.3. Realidad Virtual (RV)

La Realidad Virtual (RV) es una tecnología que crea entornos digitales simulados e interactivos, diseñados para ser experimentados e interactuados como si fueran reales. Estos entornos se generan mediante computadoras y se presentan a los usuarios a través de dispositivos como cascos de visualización (HMD) y auriculares, sumergiéndolos en mundos virtuales tridimensionales donde pueden manipular objetos virtuales utilizando controladores de movimiento u otros dispositivos de entrada[11].

2.3.1. Componentes de un Sistema de RV

Un sistema de RV típico se compone de varios elementos clave que trabajan en conjunto para crear la ilusión de realidad:

- **Dispositivos de visualización:** Cascos de realidad virtual (HMD) que presentan imágenes estereoscópicas, engañando al cerebro para que perciba profundidad y distancia, creando la sensación de estar dentro del entorno virtual.
- **Dispositivos de audio:** Auriculares o altavoces espaciales que brindan sonido envolvente, mejorando la inmersión al simular la ubicación y dirección de los sonidos en el entorno virtual.
- **Dispositivos de seguimiento:** Sensores que rastrean los movimientos del usuario, como la posición de la cabeza y las manos, y los traducen en acciones dentro del entorno virtual, permitiendo una interacción natural y fluida.
- **Software y contenido:** El software genera el entorno virtual, incluyendo modelos 3D, texturas, iluminación e interacciones. El contenido puede ser diverso, desde simulaciones científicas y educativas hasta juegos y experiencias artísticas.

2.3.2. Inmersión e Interacción

La inmersión es un componente fundamental de la RV y se refiere al grado en que un sistema de RV proyecta estímulos a los sentidos del usuario de manera extensa, envolvente, vívida e interactiva. La inmersión tiene como objetivo crear una sensación de presencia, donde los usuarios sientan que están realmente presentes en el entorno virtual.

La interacción en la RV es la comunicación entre el usuario y la aplicación de RV, mediada por dispositivos de entrada y salida. La fidelidad de interacción se refiere a la correspondencia entre las acciones físicas en el mundo virtual y las acciones físicas equivalentes en el mundo real. Existen interacciones realistas (alta fidelidad), no realistas (baja fidelidad) y mágicas (nivel medio de fidelidad).

2.3.3. Aplicaciones de la RV en la Educación

En el campo de la educación, la RV ofrece un enorme potencial para mejorar el aprendizaje al permitir a los estudiantes experimentar conceptos abstractos y complejos de una manera más tangible e interactiva.

La RV puede aprovechar las fortalezas de la percepción humana, como la percepción espacial, la percepción del tiempo y la percepción del movimiento, para crear un entorno virtual convincente y realista. Además, la incorporación de elementos de gamificación puede aumentar el compromiso y la motivación de los estudiantes, haciendo que el aprendizaje sea más divertido y efectivo.

2.4. Laboratorio Virtual

Constituyen entornos de aprendizaje interactivos que integran tecnología, pedagogía y elementos humanos para facilitar experiencias prácticas en un ambiente virtual. Su objetivo principal es introducir a los usuarios en la experimentación, resolución de problemas, deducción de resultados e interpretación científica a través de simulaciones.

2.4.1. Importancia de los Laboratorios Virtuales

Los laboratorios virtuales se han convertido en un complemento esencial en la formación experimental en ciencias, ingeniería y tecnología, gracias a los avances en tecnologías de la información y la creciente demanda de educación a distancia. Su importancia se ha visto reforzada por la pandemia de COVID-19. A continuación, se presentan las principales ventajas de estos laboratorios[12]:

- **Mejora del Aprendizaje:** Los laboratorios virtuales permiten la visualización de fenómenos complejos de manera más clara y accesible, lo que facilita la comprensión profunda y la retención de conceptos teóricos.
- **Eficiencia logística y económica:** Representan un ahorro considerable en costos de infraestructura y materiales para las instituciones educativas, lo que puede traducirse en una reducción de las tasas de matrícula. Además, eliminan la necesidad de desplazamientos físicos, lo que se traduce en un ahorro de tiempo y recursos para los estudiantes.
- **Potenciación de la experimentación:** La supervisión docente en el uso de laboratorios virtuales maximiza los resultados de aprendizaje, manteniendo el interés y el compromiso de los estudiantes al proporcionarles un entorno seguro para explorar y aprender de los errores.

Si bien los laboratorios virtuales ofrecen un amplio abanico de beneficios, es fundamental considerarlos como un complemento valioso, más no un sustituto, de los laboratorios presenciales. Estos últimos siguen siendo esenciales para el desarrollo de habilidades prácticas y manipulativas que los entornos virtuales aún no pueden replicar en su totalidad. La integración estratégica de ambas modalidades puede enriquecer significativamente la experiencia educativa y preparar a los estudiantes para los desafíos del mundo real.

2.4.2. Limitaciones en el uso de laboratorios virtuales

Simplicidad del modelo

La naturaleza virtual de los laboratorios implica una simplificación inevitable de los fenómenos y procesos reales. Esto puede conllevar una pérdida de información crucial y una distorsión de la complejidad inherente a los sistemas estudiados.

Selección y evaluación rigurosa

La idoneidad de un laboratorio virtual para una experiencia educativa específica no es universal. Se requiere una evaluación crítica por parte del docente para seleccionar la herramienta adecuada, considerando factores como los objetivos de aprendizaje, el nivel de los estudiantes y la correspondencia con los contenidos curriculares.

Competencias docentes y soporte técnico

La efectiva implementación de laboratorios virtuales exige competencias docentes en el manejo de las TIC y en la integración de estas herramientas en la pedagogía. En algunos casos, puede requerirse soporte técnico adicional para garantizar el correcto funcionamiento de las simulaciones y la resolución de problemas informáticos.

Necesidad de tutoría

La autonomía del estudiante en el uso de laboratorios virtuales puede ser limitada, requiriendo la tutoría o guía del docente para una adecuada comprensión y aprovechamiento de las simulaciones. Esto implica una inversión de tiempo y recursos por parte del educador.

Atractivo de los productos

La naturaleza digital de los resultados obtenidos en laboratorios virtuales puede restarles atractivo en comparación con los productos tangibles del laboratorio real. Esto puede afectar la motivación y el engagement de los estudiantes.

Limitación en la manipulación

Las simulaciones virtuales no permiten la manipulación directa de equipos e instrumentos de laboratorio, lo que puede ser una desventaja para el desarrollo de habilidades prácticas y la adquisición de destrezas procedimentales.

2.5. Modelado y Diseño

Para crear contenido, efectos y animaciones, es necesario desarrollar un software específico para modelos 3D. Estos programas se diseñan con el propósito de simular objetos del mundo real. Uno de los aspectos más importantes para dar credibilidad a las imágenes generadas por computadora, junto con la iluminación, son las texturas aplicadas a los objetos.

2.5.1. Modelado 3D

El modelado 3D ha revolucionado diversos sectores, introduciendo una nueva era de diseño y visualización. Mediante software especializado, se crean representaciones matemáticas precisas de objetos tridimensionales, denominados modelos 3D. Estos modelos poseen un gran valor en múltiples industrias:

- Industria cinematográfica, televisiva y de videojuegos
- Arquitectura, ingeniería y construcción
- Diseño de productos
- Ciencia y medicina

En la Tabla 2.1, se describe una breve comparación de las características de tres Aplicaciones para el modelado 3D, considerando el sistema donde corre, el costo de la licencia (Pesos mexicanos, precio por año), lenguaje de programación y su documentación.

Software	Sistema Operativo			Licencia	Lenguajes de Programación					Plataformas Compatibles		Documentación
	Windows	MacOS	Linux		C	C++	C#	Python	MEL	Unity	Unreal	
Autodesk Maya	✓	✓	✓	§21,041.00	✗	✓	✓	✓	✓	✓	✓	[13]
Blender	✓	✓	✓	Gratis	✓	✓	✗	✓	✗	✓	✓	[14]
Zbrush	✓	✓	✗	§5,992.82	✓	✓	✗	✓	✗	✓	✓	[15]

Tabla 2.1: Comparativa de los diferentes softwares para el modelado 3D

2.5.2. Modelado 2D

El modelado 2D se refiere a imágenes compuestas por dos dimensiones: ancho y largo, sin profundidad. Se emplean herramientas con entidades geométricas vectoriales como puntos, líneas, arcos y polígonos. En ámbitos como diseño de logos, tipografías e ilustraciones, el diseño bidimensional es esencial. Las animaciones 2D son planas, aunque a veces parecen tener profundidad con el uso de luz y sombra, generalmente limitada al segundo plano.

Tras la definición geométrica de los elementos que componen una escena, aplicamos propiedades a los objetos que le aportan el color, la rugosidad, brillo etc. (Material). La textura es una imagen sobrepuerta (mapeada) sobre una forma geométrica. Usada e para añadir color al elemento, o para modificar las propiedades: La reflectividad, opacidad (transparencia) o relieve (bump mapping), etc.

Software	Sistema Operativo			Licencia	Lenguajes de Programación				Tipos de Archivo				Documentación	
	Windows	MacOS	Linux		C++	JS	Python	Scheme	SVG	PNG	JPEG	BMP	AI	
Adobe Illustrator	✓	✓	✗	§3,588.00	✗	✓	✗	✗	✓	✓	✓	✓	✓	[16]
Gimp	✓	✓	✓	Gratis	✓	✗	✓	✓	✓	✓	✓	✓	✗	[17]
Inkscape	✓	✓	✓	Gratis	✓	✗	✓	✗	✓	✓	✓	✓	✓	[18]

Tabla 2.2: Comparativa de los diferentes softwares para el modelado 2D

CAPÍTULO 3

Análisis General

3.1. Metodología

Para la creación de este trabajo se optó por el modelo iterativo y creciente, dada la complejidad y amplitud del proyecto. Este enfoque permite abordar gradualmente la complejidad técnica, dividir el proyecto en etapas manejables y promover un enfoque centrado en el usuario, integrando retroalimentación continua para garantizar la satisfacción de los usuarios y optimizar la experiencia de aprendizaje. Además, facilita la adaptación a cambios y nuevos requisitos, asegurando una gestión eficaz del riesgo al identificar y mitigar problemas tempranamente, y enfatiza la validación y mejora continua del producto mediante pruebas y evaluaciones en cada iteración, asegurando la calidad y solidez del simulador final.

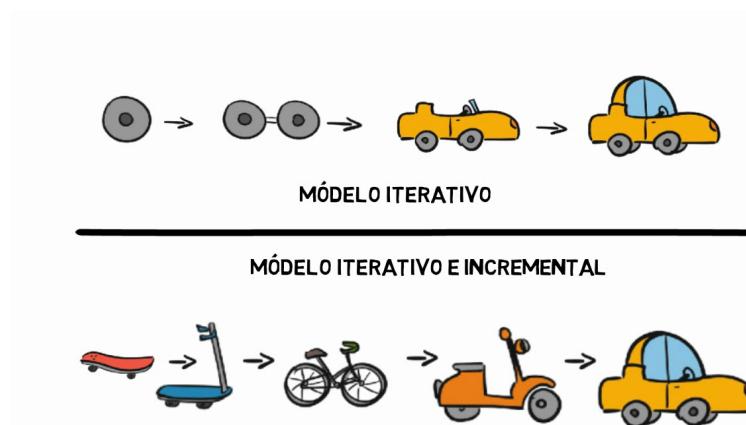


Figura 3.1: Representación del modelo iterativo y modelo iterativo e incremental

Fase Preliminar: Diseño y Planificación (Iteración 0)

- **Definición Detallada de Experimentos:** Elaboración de protocolos experimentales para los cuatro experimentos seleccionados (Reloj de Yodo, Sol de Fósforo, Síntesis de Cloruro de Amonio, Reacción Bromo-Aluminio), especificando reactivos, cantidades, procedimientos y resultados esperados.
- **Configuración del Entorno de Desarrollo:** Instalación y configuración de Unity (motor de videojuego), Blender (modelado 3D), Gimp (edición de imágenes) y entorno de programación C#. Familiarización con las herramientas.

- **Definición del Alcance del Proyecto:** Establecimiento preciso de objetivos, funcionalidades, público objetivo, restricciones y plataforma de destino (Oculus Quest 2 y Meta Quest Pro).
- **Integración del SDK de Oculus:** Configuración del proyecto Unity para realidad virtual, integración del SDK de Oculus y pruebas de compatibilidad iniciales.
- **Implementación Básica del Seguimiento de Manos:** Habilitación y configuración inicial del sistema de seguimiento de manos en Unity, con pruebas preliminares de interacción

Iteración 1: Base del Simulador y Tabla Periódica Interactiva

- **Desarrollo del Entorno Virtual:** Creación del escenario del laboratorio y configuración de iluminación y ambiente.
- **Tabla Periódica Interactiva:** Diseño e implementación de una interfaz de tabla periódica funcional, con selección de elementos, visualización de información básica (nombre, símbolo, número atómico, masa atómica) y modelos 3D interactivos de isótopos.
- **Modelado 3D de Isótopos:** Diseño y creación de modelos 3D de los isótopos principales de cada elemento.
- **Integración de Modelos 3D:** Importación y configuración de los modelos 3D en Unity, estableciendo la conexión con la tabla periódica interactiva.
- **Modelado de Materiales y Herramientas:** Diseño y modelado 3D de materiales y herramientas de laboratorio (matraces, tubos de ensayo, mecheros, etc.) con funcionalidad básica de manipulación.

Iteración 2: Tutorial Interactivo

- **Diseño del Tutorial:** Elaboración de un guion detallado con objetivos de aprendizaje, pasos secuenciales e interacciones del usuario. Diseño de elementos visuales (flechas, indicaciones, animaciones) para guiar al usuario.
- **Implementación del Tutorial:** Programación de la lógica del tutorial en Unity, incluyendo instrucciones de voz (grabadas) y manejo de interacciones con objetos virtuales.
- **Pruebas y Ajustes:** Evaluación del tutorial con usuarios, realizando ajustes en base a la retroalimentación recibida para garantizar claridad, eficacia y usabilidad.

Iteraciones 3-6: Desarrollo de Experimentos Para cada uno de los cuatro experimentos (De Básico a Ácido, La lámpara de magnesio, Nieve Química, La Bruja de Bromo), se seguirá el siguiente proceso iterativo:

- **Modelado 3D:** Creación de modelos detallados de reactivos y productos específicos del experimento.
- **Animaciones:** Desarrollo de animaciones realistas para representar las reacciones químicas y los cambios físicos observados.
- **Lógica del Experimento:** Implementación de la secuencia de pasos guiados, simulación de reacciones químicas (incluyendo cálculos de concentraciones, tiempos de reacción y cambios de color), evaluación formativa del desempeño del usuario y retroalimentación detallada.
- **Pruebas:** Pruebas del experimento para asegurar la precisión científica, la funcionalidad y la usabilidad.

Actividad	Ene.	Feb.	Mar.	Abr.	May.	Jun.	Jul.	Ago.	Sep.	Oct.	Nov.	Dic.
Análisis y requisitos del sistema												
Modelado e integración de tabla periódica												
Desarrollo del entorno virtual												
Diseño de la interfaz de usuario												
Evaluación de TT I												
Desarrollo de algoritmos de simulación												
Desarrollo de ejercicios												
Pruebas												
Reingeniería												
Redacción de Reporte Técnico												
Redacción de Manual de Usuario												
Evaluación TT II												

Figura 3.2: Cronograma de Actividades

3.2. Análisis de Requerimientos

3.2.1. Requerimientos Funcionales

RF1. *Menú Principal:* El sistema presentará un menú principal intuitivo y organizado al inicio de la ejecución. Este menú proporcionará acceso a las funcionalidades del sistema, incluyendo un tutorial, selección de experimentos.

RF2. *Selección de Elementos*

RF2.1. *Tabla Periódica Interactiva:* El sistema incorporará una tabla periódica interactiva que permita al usuario visualizar todos los elementos químicos de forma organizada.

RF2.2. *Selección Visual de Elementos:* El usuario podrá seleccionar elementos directamente desde la tabla periódica interactiva con un toque.

RF2.3. *Visualización de Información Detallada:* Al seleccionar un elemento, se mostrará una ventana emergente con información detallada sobre el mismo, incluyendo su modelo atómico en 3D, descripción química y propiedades físicas y químicas.

RF3. Experimentación Virtual

RF3.1. *Instrucciones Paso a Paso:* El sistema guiará al usuario a través de cada experimento mediante instrucciones claras y concisas que detallen la cantidad de reactivos a utilizar y el procedimiento a seguir.

RF3.2. *Simulación de Reacciones:* El sistema simulará la reacción química entre los elementos seleccionados, siguiendo principios químicos establecidos.

RF3.3. *Observación de Efectos:* El usuario podrá observar los efectos de la reacción química en el entorno virtual, como la formación de nuevos compuestos, la liberación de gases o cambios de color.

3.2.2. Requerimientos No Funcionales

RNF1. Rendimiento

RNF1.1. *Fluidez Gráfica:* El sistema debe exhibir una tasa de cuadros por segundo (FPS) constante y superior a 60 FPS, para garantizar una experiencia de usuario inmersiva y sin interrupciones.

RNF1.2. *Tiempo de Respuesta:* El sistema debe presentar un tiempo de respuesta mínimo a las acciones del usuario, minimizando los retrasos y garantizando una interacción dinámica y receptiva.

RNF2. Usabilidad

RNF2.1. *Interfaz Intuitiva:* La interfaz de usuario debe seguir principios de diseño intuitivos y accesibles, utilizando elementos visuales claros, jerarquías de información bien definidas y una navegación sencilla.

RNF2.2. *Navegación Sencilla:* La interfaz de usuario debe seguir principios de diseño intuitivos y accesibles, utilizando elementos visuales claros, jerarquías de información bien definidas y una navegación sencilla.

RNF2.3. *Retroalimentación Clara:* El sistema debe proporcionar retroalimentación clara y concisa al usuario sobre sus acciones e interacciones dentro del entorno virtual.

3.2.3. Reglas de Negocio

El prototipo de simulador de laboratorio de química inorgánica en realidad virtual se regirá por las siguientes reglas de negocio:

RN-01. *Selección de Elementos:*

- Un elemento químico se considera seleccionado cuando el usuario toca su ficha en la tabla periódica con su dedo índice.
- Solo se puede seleccionar un elemento a la vez. Si se selecciona un elemento que ya existe en la escena, este se destruye y se crea uno nuevo en la zona designada.
- La GUI se actualizará con la información del elemento seleccionado.

RN-02. *Creación de Compuestos:*

- Para crear un compuesto, el usuario debe llevar los elementos necesarios a la zona de creación.
- El sistema verificará si cada elemento es correcto para el compuesto en dos ocasiones: al seleccionarlo y al entrar en la zona de creación.
- Si un elemento es incorrecto, el sistema notificará al usuario mediante un mensaje de voz y texto, y le indicará que lo lleve a la zona de desecho.
- El compuesto se formará solo cuando todos los elementos necesarios estén presentes en la zona de creación.
- Una vez creado el compuesto, se mostrará su modelo 3D en el área de experimentación y se actualizará la GUI correspondiente.

RN-03. *Interacción con Objetos:*

- El usuario puede agarrar, mover y soltar compuestos, instrumental y elementos utilizando el seguimiento de manos (hand tracking).
- La interacción con objetos estará limitada a las acciones permitidas en cada etapa del experimento.

RN-04. *Simulación de Reacciones:*

- La simulación de reacciones químicas presentará efectos visuales básicos, como cambios de color y liberación de gases, priorizando la comprensión de los conceptos fundamentales sobre el realismo detallado.

RN-05. *Etapas del Experimento:*

- Cada experimento se dividirá en cinco etapas: presentación, balanceo de ecuación, selección de elementos y creación de compuestos, desarrollo y simulación, y explicación de resultados y conclusiones.
- El usuario avanzará a la siguiente etapa solo después de completar exitosamente la etapa anterior.
- En la etapa de desarrollo y simulación, el usuario seguirá instrucciones paso a paso y realizará las acciones necesarias para completar el experimento.

RN-06. Retroalimentación al Usuario:

- El sistema proporcionará retroalimentación visual y auditiva al usuario en cada etapa del experimento.
- En caso de errores, se mostrarán mensajes claros y concisos en la GUI indicando el problema y cómo solucionarlo.

RN-07. Navegación:

- Al finalizar un experimento, el usuario será redirigido automáticamente al escenario de bienvenida.
- Desde el escenario de bienvenida, el usuario podrá seleccionar otro experimento o repetir el tutorial.

RN-08. Eliminación de Elementos:

- Los elementos químicos y compuestos pueden ser eliminados del área de trabajo arrastrándolos a la zona de desecho.
- El sistema verificará si el elemento o compuesto puede ser desecharado en ese momento del experimento.

RN-09. Balanceo de Ecuaciones:

- En la etapa de balanceo de ecuaciones, el usuario deberá ajustar los coeficientes estequiométricos de la reacción química para cumplir con la ley de conservación de la masa.
- El sistema verificará si la ecuación está correctamente balanceada antes de permitir al usuario avanzar a la siguiente etapa.

RN-010. Tutorial Interactivo:

- El tutorial guiará al usuario a través de las funciones básicas del simulador, incluyendo la selección de elementos, la creación de compuestos y la interacción con el instrumental.
- El tutorial será opcional y podrá ser repetido en cualquier momento desde el menú principal.

3.2.4. ¿Como debe venderse?

Como una herramienta educativa y de entretenimiento inmersiva que permite a los usuarios explorar y experimentar con la química inorgánica de manera segura y divertida. El simulador ofrece una experiencia interactiva y visualmente atractiva, que permite a los usuarios manipular elementos químicos, realizar experimentos guiados y observar reacciones en tiempo real en un entorno de laboratorio virtual.

Se puede distribuir de dos formas:

- **Versión demo gratuita:** permitiendo a los usuarios potenciales experimentar con una selección limitada de experimentos y familiarizarse con la interfaz y el funcionamiento del simulador.
- **Licencia completa:** proporcionando acceso completo a todas las funcionalidades y experimentos disponibles en el simulador.

El público objetivo del simulador incluye:

- Estudiantes de secundaria y nivel medio superior que cursan química.
- Profesores de química que buscan herramientas innovadoras para sus clases.
- Aficionados a la ciencia y entusiastas de la realidad virtual.
- Museos y centros de ciencia que buscan ofrecer experiencias educativas interactivas. virtual.

3.2.5. Limitaciones del Sistema

El prototipo de simulador de laboratorio de química inorgánica en realidad virtual presentará las siguientes limitaciones inherentes a su diseño y alcance:

- **Ámbito Exclusivo en Química Inorgánica:** El simulador se circunscribe a la simulación de experimentos dentro del dominio de la química inorgánica, excluyendo otras ramas de la química, como la orgánica, la bioquímica o la fisicoquímica.
- **Experimentación Guiada y Estructurada:** La experiencia del usuario se centra en experimentos predefinidos y guiados, priorizando la facilidad de uso y la comprensión de los conceptos fundamentales de la química inorgánica. No se contempla la posibilidad de realizar experimentos libres o personalizados en esta versión.
- **Limitaciones en el Realismo:** Aunque se busca la mayor fidelidad posible a la realidad, es importante destacar que el simulador ofrece una representación virtual de los fenómenos químicos. Algunas reacciones o propiedades químicas podrían ser simplificadas o idealizadas para facilitar la comprensión y el uso educativo, sin comprometer la validez de los conceptos aprendidos.
- **Plataformas Compatibles:** El simulador será compatible exclusivamente con los dispositivos de realidad virtual Oculus Meta Quest 2 y Oculus Meta Quest Pro.

3.3. Tecnologías de Desarrollo

3.3.1. Modelado 3D y Animación

3.3.1.1. Blender

Blender 3D es un software multiplataforma y gratuito especializado en el modelado, iluminación, renderizado, animación y creación de modelos 3D. También incluye composición digital utilizando nodos, edición de video, escultura y pintura digital.

El objetivo principal de Blender es generar imágenes 2D a partir de escenas 3D, utilizando diversos motores gráficos, tanto de pre-renderizado como de tiempo real, que vienen integrados por defecto. El programa ofrece una variedad de figuras geométricas primitivas, como curvas, mallas poligonales, vacíos y metaballs, además de una amplia gama de herramientas para modelado, renderizado, rigging, edición de video, audio, composición y animación.

Blender también incluye herramientas de animación avanzadas como cinemática inversa, deformaciones con armaduras o cuadrículas, vértices de carga y partículas estáticas. Además, proporciona características interactivas para el desarrollo de juegos, incluyendo detección de colisiones, simulaciones dinámicas, aplicación de físicas y lógica.

La interfaz gráfica OpenGL de Blender es uniforme y personalizable en las principales plataformas, y permite el uso de scripts en Python para automatizar y controlar varias tareas. Admite formatos gráficos como TGA, JPG, Iris, SGI y TIFF, e incluye simulaciones dinámicas para cuerpos blandos, partículas y fluidos. También cuenta con un sistema de partículas estáticas para simular cabello y pelaje, con opciones avanzadas de shaders para texturas realistas.

Desde su creación en 2001, Blender ha sido reconocido por su constante desarrollo y mejoras, proporcionando a los usuarios una experiencia única y en evolución.

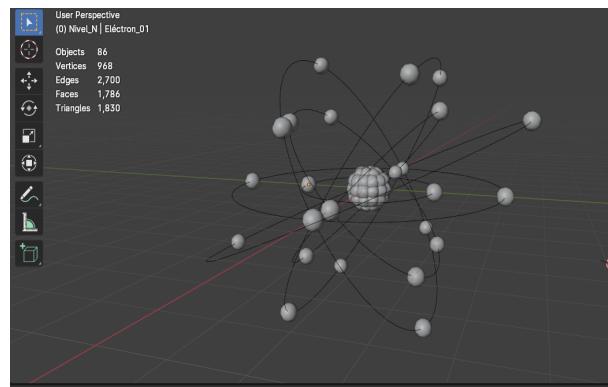


Figura 3.3: Modelo de un átomo en Blender

3.3.1.2. Gimp

Este software de manipulación de imágenes ha experimentado una notable evolución a lo largo del tiempo. Utiliza GTK como biblioteca de controles gráficos.

Permite trabajar con imágenes en capas, lo que facilita la modificación independiente de cada objeto en la imagen. Las capas pueden organizarse en una pila, subiendo o bajando de nivel

para facilitar la edición. Cada capa tiene su propia visibilidad y nivel de transparencia, y hay numerosas formas de combinar las relaciones entre ellas.

Además, es posible producir imágenes de manera completamente automatizada y no interactiva, así como realizar un procesamiento por lotes que cambia colores o convierte series de imágenes. Para tareas automatizadas más simples, se recomienda utilizar un paquete como ImageMagick.

Imágenes

Una imagen en GIMP puede ser más compleja de lo que parece. No es solo un dibujo en un lienzo, sino una pila de capas, una máscara de selección, un conjunto de canales y rutas.

Cuenta con un sistema de gestión de memoria avanzado basado en bloques de píxeles, lo que permite manejar imágenes muy grandes sin dificultad.

Capas

Una imagen con capas es similar a un fajo de papeles transparentes apilados. Se puede dibujar en cada papel y ver el contenido de otras hojas a través de las áreas transparentes. También es posible mover una hoja en relación con las demás.

Una imagen en GIMP puede contener muchas capas, incluso docenas. Las capas no tienen que ser opacas ni cubrir toda la extensión de la imagen, por lo que se puede ver más que la capa superior, observando elementos de otras capas.

Canales

Los canales son útiles cuando se trabaja en una imagen que necesita ajustes en un color específico. Ver los canales como máscaras permite restringir la salida del color que el canal representa. Usando filtros en la información del canal, se pueden crear muchos efectos variados y sutiles sobre una imagen. Con estos canales, es posible crear otros canales, conocidos como máscaras de canal.

3.3.2. Motor de Videojuegos

3.3.2.1. Unity

Unity es una plataforma líder en el desarrollo de juegos, compatible con múltiples plataformas, ampliamente utilizada para crear experiencias interactivas tanto en 3D como en 2D. Su ecosistema completo, que incluye un amplio conjunto de herramientas y una comunidad activa, facilita la creación de simulaciones de realidad virtual realistas, la interacción natural con elementos virtuales y el desarrollo de interfaces de usuario intuitivas.

El entorno integrado de Unity está compuesto por diversas partes y componentes que permiten el desarrollo de simuladores. Está desarrollado en C#, un lenguaje de programación multiparadigma que, al ser de acceso libre, fomenta la creación de contenido por parte de la comunidad y la creación de sus propias librerías.

C# es la tercera versión de C y una mejora respecto a C++, lo que simplifica la exportación y comunicación con programas escritos en estos lenguajes sin necesidad de una interfaz compleja.

3.3.2.2. C#

C# es un lenguaje de programación que se centra en componentes, está orientado a objetos y garantiza la seguridad de tipos. Esto permite a los desarrolladores crear una variedad de aplicaciones seguras y robustas que se ejecutan en el entorno .NET. Es especialmente adecuado para la creación y utilización de componentes de software, y ofrece características que respaldan nuevas cargas de trabajo y prácticas de diseño de software en evolución. En esencia, C# es un lenguaje orientado a objetos.

3.3.2.3. GameObject

Los GameObject son elementos esenciales en Unity, ya que pueden representar diversos elementos como avatares, objetos o el escenario mismo. Por sí solos, estos elementos no realizan ninguna función, pero actúan como contenedores para sus Componentes.

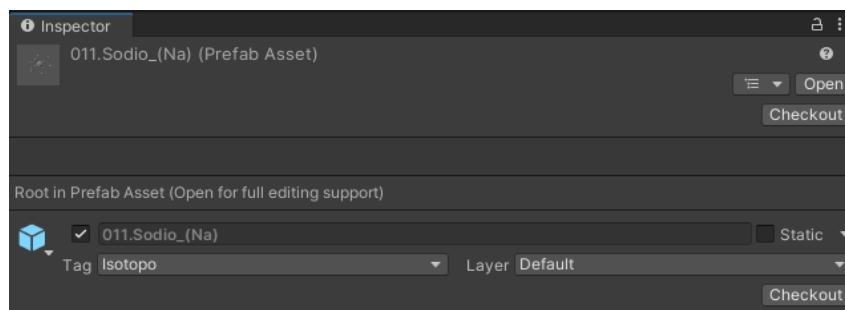


Figura 3.4: Game Object en Unity

3.3.2.4. Componentes

Los componentes son los que definen el comportamiento de cada GameObject en el juego. No existe un límite establecido en cuanto a la cantidad de componentes que puede tener un objeto.

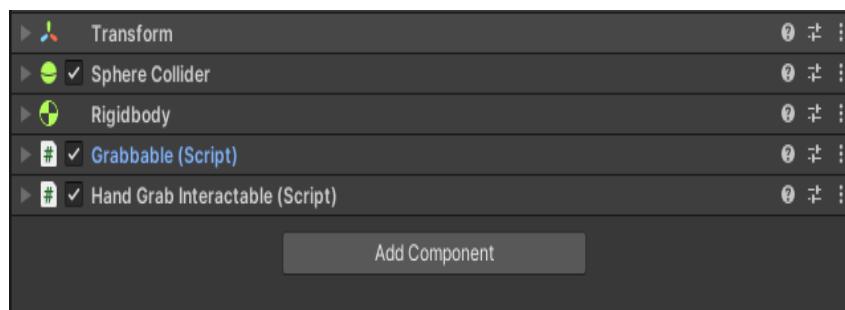


Figura 3.5: Componentes de un Elemento en Unity

3.3.2.5. Script

El Script es un componente diseñado por el usuario para realizar una acción específica en el juego. Consiste en una secuencia de líneas de comandos escritas principalmente en C# o en UnityScript. Sin embargo, otros lenguajes .NET pueden ser utilizados con Unity si son compilados en un DLL compatible.

Los Scripts permiten activar eventos en su juego, modificar propiedades del Componente en tiempo real y responder al input del usuario de la manera que desee. Puede crear un nuevo script desde el menú “Create” en la parte superior izquierda del panel del Proyecto, o seleccionando Assets > Create > C# Script desde el menú principal.

Cuando se adjunta un componente script a un GameObject, se crea una nueva instancia del objeto definido por el plano. Es importante tener en cuenta que un script solo define un plano para un componente; su código no se activará hasta que una instancia del script sea adjuntada al GameObject.

3.3.2.6. ScriptableObject

ScriptableObject es una clase en Unity que permite almacenar grandes cantidades de datos compartidos independientes de las instancias de script. A diferencia de SerializableObject, esta clase está diseñada específicamente para este propósito y no debe confundirse con ella.

Cuando se utiliza ScriptableObject, los datos se almacenan por referencia en lugar de por valor, lo que significa que las instancias de ScriptableObject mantienen los datos y las instancias de script solo contienen referencias a estos datos. Esto ayuda a reducir el uso de memoria al evitar la duplicación de valores.

Un caso de uso común para ScriptableObject es definir conjuntos de datos conectables, como en el caso de una tienda NPC en un juego RPG, donde cada zona puede ofrecer diferentes tipos de ítems para la venta. Los objetos de ScriptableObject personalizados pueden definir estos conjuntos de datos, lo que permite una fácil configuración y personalización del juego.

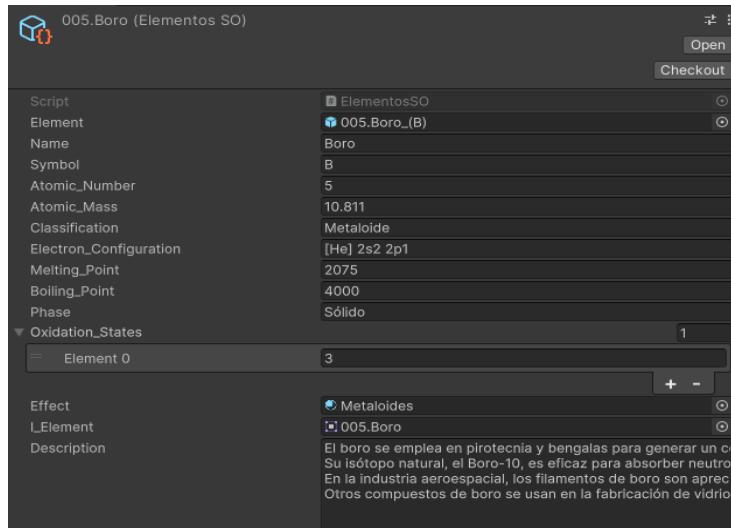


Figura 3.6: ScriptableObject de un Elemento en Unity

3.3.3. Integración de Oculus Meta Quest 2 y Oculus Meta Quest Pro

Meta Quest 2 y Meta Quest Pro son dispositivos de realidad virtual (VR) desarrollados por Meta Platforms (anteriormente Facebook), cada uno diseñado para satisfacer necesidades y expectativas específicas de los usuarios.

Meta Quest 2: Lanzado en 2020, este dispositivo autónomo ofrece una experiencia de VR inmersiva sin necesidad de una PC externa, proporcionando un rendimiento fluido en la mayoría de aplicaciones y juegos disponibles en su plataforma, brindando una experiencia visual de alta calidad.



Figura 3.7: Oculus Meta Quest 2

Meta Quest Pro: Lanzado en 2022, este dispositivo de gama alta está orientado a profesionales y entusiastas de la VR, ofrece un rendimiento superior y capacidades avanzadas. Además, incorpora seguimiento ocular y facial, así como cámaras de alta resolución, que permiten una interacción más natural y expresiva en entornos virtuales, y habilitan funciones de realidad mixta.



Figura 3.8: Oculus Meta Quest Pro

SDK de Oculus

El SDK de Oculus proporciona las herramientas y bibliotecas necesarias para integrar el simulador con las gafas VR Oculus Meta Quest 2 y Oculus Meta Quest Pro de manera nativa. Permite el seguimiento preciso de la cabeza y el movimiento de las manos, la representación 3D estereoscópica de alta calidad y la interacción natural con elementos virtuales, creando una experiencia de realidad virtual inmersiva y atractiva.

Característica	Meta Quest 2	Meta Quest Pro
Resolución por Ojo	1832x1920 pixeles	1800x1920 pixeles
Frecuencia de Actualización	72-90 Hz	72-90 Hz
Procesador	Qualcomm Snapdragon XR2	Qualcomm Snapdragon XR2+
RAM	6 GB	12 GB
Almacenamiento	128 GB / 256 GB	256 GB
Seguimiento Ocular y Facial	✗	✓
Cámaras	4	10
Realidad Mixta	✗	✓
Duración de la batería	2-3 horas	2-3 horas
Precio aproximado	\$299 USD (128 GB) / \$399 USD (256 GB)	\$999 USD

Tabla 3.1: Comparación de especificaciones entre Oculus Meta Quest 2 y Oculus Meta Quest Pro

3.3.4. Handtracking

El seguimiento de manos (hand tracking) es una tecnología que permite la detección y seguimiento en tiempo real de los movimientos de las manos del usuario a través de dispositivos equipados con cámaras, como las gafas de realidad virtual Meta Quest. Esta tecnología se basa en el análisis de imágenes capturadas por las cámaras, identificando puntos clave en las manos y rastreando su movimiento a medida que el usuario las desplaza. El seguimiento de manos ofrece una modalidad de interacción más natural e intuitiva con dispositivos y entornos virtuales, eliminando la necesidad de controladores físicos.

En Realidad virtual (VR) y realidad mixta: Permite una interacción más inmersiva y realista con entornos virtuales, mejorando la experiencia del usuario en juegos, simulaciones y aplicaciones de entrenamiento.

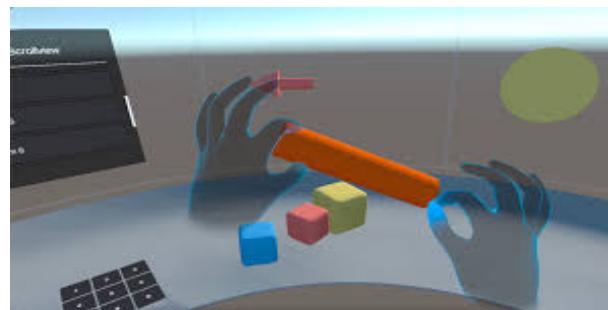


Figura 3.9: Ejemplo de Handtracking

3.4. Análisis Económico

3.4.1. Análisis por Puntos de Función

Métrica de punto de función La métrica de punto de función (PF) puede usarse de manera efectiva como medio para medir la funcionalidad que entra a un sistema.

Para obtener el PF vamos a considerar los valores de dominio de información que se definen de la siguiente forma:

Número de entradas externas (EE): 5

- Elegir experimento/tutorial (Simple)
- Seleccionar elemento (Simple)
- Eliminar elemento (Simple)
- Crear compuesto (Simple)
- Interactuar con instrumental (Simple)

Número de salidas externas (SE): 3

- Visualización del experimento (Complejo)
- Resultados del experimento (Complejo)
- Visualización del Entorno de Laboratorio (Promedio)

- Visualización de Elementos y Compuestos (Promedio)
- Indicaciones visuales y auditivas (Promedio)

Número de consultas externas (CE): 0

Número de archivos lógicos internos (ALI): 3

- Datos de experimentos (Promedio)
- Datos de elementos (Promedio)
- Datos de compuestos (Promedio)

Número de archivos de interfaz externos (AIE): 0

Valor de dominio de información	Factor Ponderado					Total
	Conteo	Simple	Promedio	Complejo		
Entradas Externas (EE)	5 Simples	*	3	4	6	= 15
Salidas Externas (SE)	2 Complejos, 3 Promedio	*	4	5	7	= 29
Consultas Externas (CE)	0	*	3	4	6	= 0
Archivos Lógicos Internos (ALI)	3 Promedios	*	7	10	15	= 30
Archivos de Interfaz Externos (AIE)	0	*	5	7	10	= 0
Conteo Total						74

Tabla 3.2: Cálculo de Puntos de Función (PF)

Una vez obtenido lo anterior continuaremos con la obtención de los F_i ($i = 1$ a 14) son factores de ajuste de valor (FAV) con base en respuestas a las siguientes preguntas cada una de estas preguntas se responde usando una escala que varía de 0 (no importante o aplicable) a 5 (absolutamente esencial).

1. *¿Requiere el sistema copias de seguridad y recuperación fiables?* 0 no influye en el sistema.
2. *¿Se requiere comunicación de datos?* 0 no influye en el sistema.
3. *¿Existen funciones de procesamiento distribuido?* 0 no influye en el sistema.
4. *¿Es crítico el rendimiento?* 5 es absolutamente esencial
5. *¿Se ejecutará en un entorno operativo existente y fuertemente utilizado?* 2 de moderada importancia
6. *¿Requiere entrada de datos interactiva?* 5 es absolutamente esencial
7. *¿Requiere la entrada interactiva transacciones en múltiples pantallas?* 0 no influye en el sistema.
8. *¿Se actualizan los archivos maestros de forma interactiva?* 2 de moderada importancia
9. *¿Son complejas las entradas, salidas, archivos o consultas?* 4 significativa importancia
10. *¿Es complejo el procesamiento interno?* 4 significativa importancia
11. *¿Se ha diseñado el código para ser reutilizable?* 5 es absolutamente esencial
12. *¿Están incluidas en el diseño la conversión y la instalación?* 1 de poca importancia
13. *¿Se ha diseñado para múltiples instalaciones en diferentes organizaciones?* 2 de moderada importancia
14. *¿Se ha diseñado para facilitar cambios y ser fácil de usar?* 4 significativa importancia

Una vez terminado de responder las preguntas realizamos las sumas de los F_i :

$$\Sigma(F_i) = 34$$

Teniendo ya estos datos podemos obtener los puntos de función utilizando la siguiente expresión

$$PF = \text{Conteo Total} * (0,65 + 0,01 * \Sigma(F_i))$$

$$PF = 74 * (0,65 + 0,01 * 34)$$

$$PF = 73,26$$

Estimación COCOMO

El modelo COCOMO, desarrollado por Barry W. Boehm, estima la cantidad de meses-hombre necesarios para desarrollar un software. Consta de tres submodelos (orgánico, semiacoplado y empotrado) que varían en detalle y precisión. La elección del submodelo depende de la cantidad estimada de miles de líneas de código (KLDC), que se puede aproximar utilizando el factor de productividad (PF) obtenido previamente y la siguiente tabla de referencia.

Language	QSM SLOC/FP Data			
	Avg	Median ↑	Low	High
Brio +	14	14	13	16
Perl *	24	15	15	60
ABAP (SAP) *	28	18	16	60
Cross System Products (CSP) +	20	18	10	38
Lotus Notes *	23	21	19	40
SQL *	21	21	13	37
Cool:Gen/IEF *	32	24	10	82
Powerbuilder *	26	28	7	40
LINC II	29	30	22	38
PACBASE *	35	32	22	60
Natural *	40	34	34	53
FoxPro	36	35	34	38
PL/SQL *	37	35	13	60
SAS *	38	37	22	55
HTML *	34	40	14	48
Oracle *	37	40	17	60
Cognos Impromptu Scripts +	47	42	30	100
Visual Basic *	42	44	20	60
Focus *	43	45	45	45
JCL *	62	48	25	221
J2EE *	46	49	15	67
C++ *	50	53	25	80
Java *	53	53	14	134
JavaScript *	47	53	31	63
ASP*	51	54	15	69
COBOL *	61	55	23	297
C# *	54	59	29	70
.NET *	57	60	53	60
Siebel *	59	60	51	60
VB.NET *	52	60	26	60
Datastage	71	65	31	157
Sabretalk *	70	66	45	109
SLOGAN *	75	75	74	75
PL/I *	64	80	16	80
REXX *	77	80	50	80
Assembler *	119	98	25	320
C *	97	99	39	333
Excel *	209	191	131	315

Figura 3.10: Tabla de lenguajes de puntos de función - QSM[19]

El lenguaje utilizado será C#, que tiene un LOC promedio de 54 líneas por función. Con estos valores, se aplican a la siguiente fórmula para obtener una estimación en miles de líneas de código:

$$KLDC = (PF) * (\text{Promedio de líneas por función})$$

$$KLDC = 73,26 * 54$$

$$KLDC = 3,956,04$$

$$\boxed{KLDC = 3.9560 \text{ miles de líneas de código}}$$

A continuación, se observa en cuál submodelo se clasificará el presente proyecto según los siguientes criterios:

- Orgánico: Proyectos pequeños con menos de 50 KLCD
- Semiacoplado: Proyectos de complejidad media con menos de 300 KLCD
- Empotrado: Proyectos muy complejos donde no hay experiencia previa y se utiliza tecnología de vanguardia

Dado que el proyecto tiene menos de 50 KLCD, se clasifica en el submodelo Orgánico. Aunque todos los submodelos utilizan las mismas ecuaciones para la estimación, emplean diferentes coeficientes, los cuales se presentan en la siguiente tabla:

Proyecto de Software	a	b	c	d
Orgánico	2.4	1.05	2.5	0.38
Semiacoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Tabla 3.3: COCOMO básico

Se procede a realizar el cálculo del esfuerzo por persona-mes. Siendo a y b coeficientes de la Tabla 3.3

$$E = a * (KLDC)^b$$

$$E = (2,4) * (3,9560)^{1,05}$$

$$E = 10,17 \text{ persona-mes}$$

A continuación, se calcula la duración del proyecto utilizando la siguiente fórmula:

$$D = c * E^d$$

$$D = (2,5) * (10,17021979)^{0,38}$$

$$D = 6.035 \text{ meses}$$

Por lo que se puede observar los resultados arrojan que el tiempo en el que se debe desarrollar el proyecto es de 6 meses con 2 personas.

Por lo tanto, se concluye que la estimación del tiempo de desarrollo del proyecto es de 6 meses con 2 personas, lo que implica un esfuerzo de 12 meses-persona. Al considerar que el proyecto será desarrollado por una sola persona, la duración ajustada del proyecto se estima en aproximadamente 10 meses. Este tiempo es justo, pero no es tolerante a retrasos.

Es importante destacar que, al tener un margen de tiempo limitado para el desarrollo del proyecto, se presenta un riesgo moderado a significativo. Cualquier retraso en el cronograma puede afectar negativamente la implementación de algunas funcionalidades del sistema.

3.4.2. Análisis de costos

El análisis de costos del sistema se basa en una lista detallada de todos los gastos asociados a su desarrollo.

El proyecto considera un total de 660 horas-hombre de desarrollo.

Mano de obra

El costo de la mano de obra se calcula considerando un total de 660 horas-hombre, con un salario estimado de \$70.00 MXN por hora o \$11,200.00 MXN mensuales.

La distribución del tiempo de desarrollo y los costos asociados se detalla en la Tabla 3.4:

Equipo	Costo	Tiempo
Modelado 3D, 2D y VFX	\$11,550.00	165 hrs.
Software	\$27,720.00	396 hrs.
VR	\$6,930.00	99 hrs.
Total	\$46,200.00	660 hrs.

Tabla 3.4: Costos de Desarrollo

Equipo de Desarrollo

El equipo de desarrollo incluye todos los recursos necesarios para diseñar y ejecutar el prototipo. Los componentes principales son:

- Computadora de Desarrollo

Se utilizará una laptop de gama media-alta para ejecutar el sistema.

MSI Katana 15 B12V: \$25,440.00

- **Procesador:** Intel Core i7 12650H
- **Memoria RAM:** 16 GB DDR5
- **Almacenamiento:** 1 TB SSD
- **Tarjeta gráfica:** NVIDIA GeForce RTX 4070
- **Sistema operativo:** Windows 11



Figura 3.11: Laprop MSI Katana 15 B12V

- Gafas de Realidad Virtual

Se cuenta con las gafas Oculus Meta Quest Pro con un costo aproximado en México de \$20,000.00 MXN y las gafas Oculus Meta Quest 2 con un costo aproximado de \$10,000.00 MXN



(a) Oculus Meta Quest Pro



(b) Oculus Meta Quest 2

Figura 3.12: Sistemas Compatibles

Objeto	Costo
Laptop MSI Katana 15 B12V	\$25,440.00
Oculus Meta Quest Pro	\$20,000.00
Oculus Meta Quest 2	\$10,000.00
Total	\$55,440.00

Tabla 3.5: Costos de Infraestructura

Licenciamiento

Se incluyen las licencias y herramientas externas necesarias para el desarrollo, como se detalla en la Tabla 3.6.

Licenciamiento	Costo
Unity Assets	\$5,000.00
SPEECHGEN.IO	\$600.00
3D	\$500.00
Total	\$6,100.00

Tabla 3.6: Costos de Licencias

Gastos de Operación

Se estimaron gastos operativos aproximados de \$11,000.00 MXN.

El costo total del sistema se calcula sumando todos los conceptos anteriores:

$$\textbf{Costos del sistema} = \text{Mano de obra} + \text{Equipo de Desarrollo} + \text{Licenciamiento} + \text{Gastos de Operación}$$

$$\textbf{Costo del sistema} = \$46,200.00 + \$55,440.00 + \$6,100.00 + \$11,000.00$$

$$\textbf{Costo del sistema} = \$118,300.00 \text{ MXN}$$

Este costo incluye la curva de aprendizaje asociada con el desarrollo inicial. Si el proyecto se repitiera desde cero con la necesidad de adquirir nuevamente la infraestructura y las licencias, pero considerando que el tiempo requerido sería la mitad, el costo total estimado se reduciría

aproximadamente a \$95,200.00 MXN. En caso de contar ya con toda la infraestructura y las licencias necesarias, el costo total disminuiría aún más, alcanzando aproximadamente \$34,100.00 MXN.

3.5. Modelado del Sistema

3.5.1. Casos de Uso

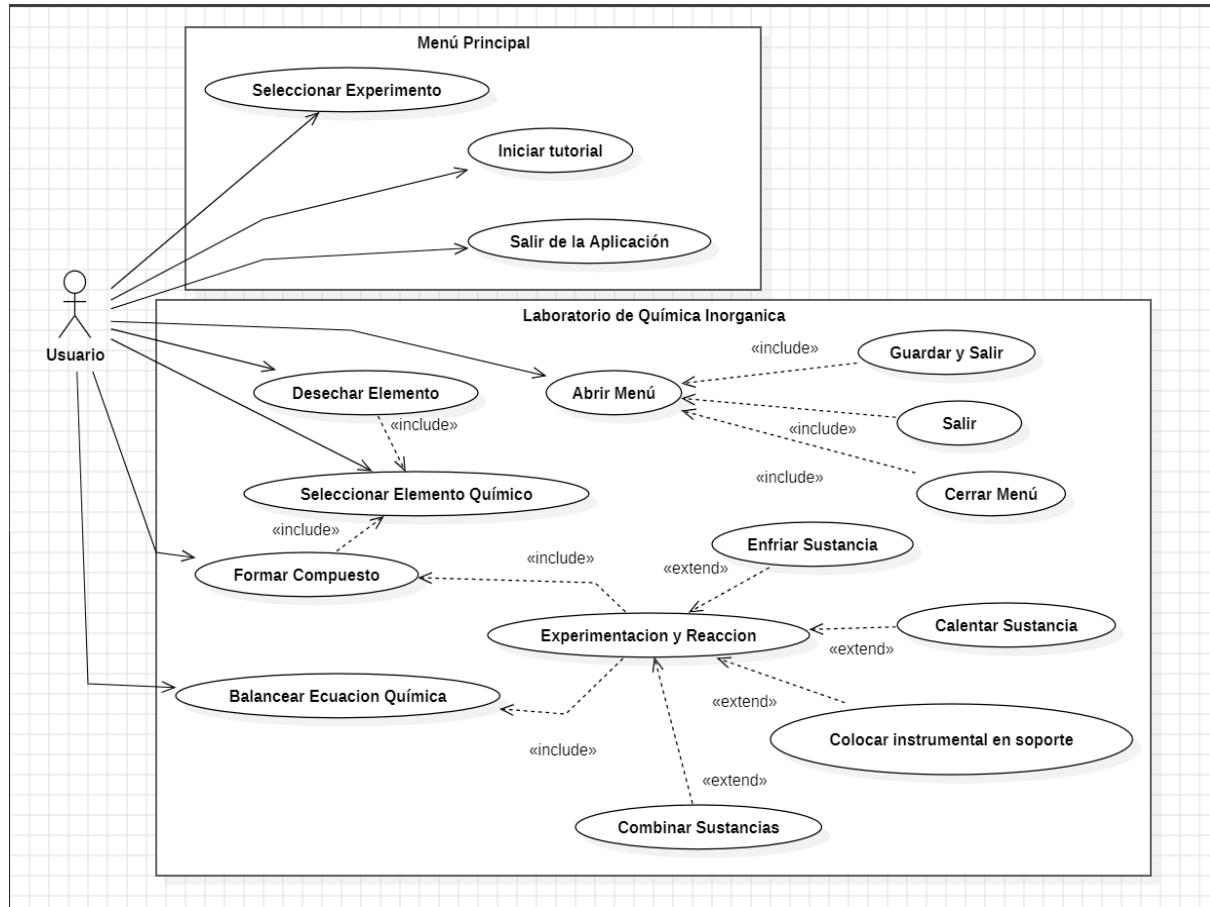


Figura 3.13: Diagrama de casos de uso para el sistema

Caso de Uso	Seleccionar Experimento
Descripción corta	El usuario elige un experimento de química para realizar en el simulador.
Precondición	El usuario se encuentra en el menú principal del simulador.
Postcondición	El sistema carga el entorno y los materiales del experimento seleccionado, y presenta al usuario las instrucciones iniciales.
Situaciones de error	Selección Inválida: El usuario intenta seleccionar un experimento que no existe o no está disponible.
Actores	Usuario
Disparadores	El usuario interactúa con la interfaz del menú principal y selecciona un experimento de la lista.
Proceso estándar	<ol style="list-style-type: none">1. El sistema muestra una lista de experimentos disponibles en el menú principal.2. El usuario utiliza el hand tracking para seleccionar un experimento de la lista.3. El sistema verifica la validez de la selección.4. El sistema carga el entorno virtual del laboratorio, incluyendo los materiales y equipos necesarios para el experimento.5. El sistema presenta al usuario las instrucciones iniciales del experimento.

Continúa en la siguiente página

Caso de Uso	Seleccionar Experimento
--------------------	-------------------------

3A. Selección Inválida:

1. El sistema detecta que la selección del usuario no es válida.
 2. El sistema muestra un mensaje de error indicando que el experimento no existe o no está disponible.
 3. El sistema regresa al menú principal.
-

Tabla 3.7: Descripción del caso de uso Seleccionar Experimento

Caso de Uso	Iniciar Tutorial
--------------------	------------------

Descripción corta	El usuario accede a un tutorial interactivo para aprender a utilizar las funciones básicas del simulador.
--------------------------	---

Precondición	El usuario se encuentra en el menú principal del simulador.
---------------------	---

Postcondición	El usuario ha completado el tutorial y está familiarizado con las interacciones básicas del simulador.
----------------------	--

Situaciones de error	No aplica.
-----------------------------	------------

Actores	Usuario
----------------	---------

Disparadores	El usuario selecciona la opción Iniciar Tutorial en el menú principal.
---------------------	--

Continúa en la siguiente página

Caso de Uso	Iniciar Tutorial
<hr/>	
Proceso estándar	<ol style="list-style-type: none"> 1. El sistema presenta una serie de instrucciones y demostraciones interactivas que guían al usuario a través de las funciones básicas del simulador, como la selección de elementos de la tabla periódica, la manipulación de objetos y la realización de mediciones. 2. El usuario sigue las instrucciones y completa las tareas interactivas utilizando el hand tracking. 3. El sistema proporciona retroalimentación visual y auditiva al usuario sobre su desempeño en cada tarea. 4. Una vez completadas todas las tareas del tutorial, el sistema regresa al menú principal.
<hr/>	
Procesos alternativos	No aplica.

Tabla 3.8: Descripción del caso de uso Iniciar Tutorial

Caso de Uso	Balancear Ecuación Química
<hr/>	
Descripción corta	El usuario balancea una ecuación química proporcionada por el sistema.
<hr/>	
Precondición	El sistema ha presentado al usuario una ecuación química desbalanceada como parte de un experimento.
<hr/>	
Postcondición	El usuario ha balanceado correctamente la ecuación química, y el sistema lo verifica y permite avanzar al siguiente paso del experimento.
<hr/>	
Situaciones de error	Balanceo Incorrecto: El usuario intenta confirmar un balanceo incorrecto de la ecuación.

Continúa en la siguiente página

Caso de Uso	Balancear Ecuación Química
--------------------	----------------------------

Actores	Usuario
----------------	---------

Disparadores	El sistema presenta una ecuación química desbalanceada y solicita al usuario que la balancee.
---------------------	---

Proceso estándar	<ol style="list-style-type: none">1. El sistema muestra la ecuación química desbalanceada en la interfaz de usuario.2. El usuario utiliza el hand tracking para interactuar con los coeficientes de la ecuación, incrementándolos o decrementándolos según sea necesario.3. El usuario confirma que ha terminado de balancear la ecuación.4. El sistema verifica si la ecuación está balanceada correctamente.5. Si la ecuación está balanceada, el sistema permite al usuario avanzar al siguiente paso del experimento.
-------------------------	---

Procesos alternativos	<p>4A. Balanceo Incorrecto:</p> <ul style="list-style-type: none">■ El sistema detecta que la ecuación no está balanceada correctamente.■ El sistema muestra un mensaje de error indicando que el balanceo es incorrecto.■ El sistema permite al usuario seguir intentando balancear la ecuación.
------------------------------	--

Tabla 3.9: Descripción del caso de uso Balancear Ecuación Química

Caso de Uso	Seleccionar Elemento Químico
Descripción corta	El usuario selecciona un elemento químico de la tabla periódica interactiva.
Precondición	La tabla periódica interactiva está visible en la interfaz del usuario.
Postcondición	<ul style="list-style-type: none">■ El elemento seleccionado se resalta visualmente en la tabla periódica.■ Se muestra información detallada sobre el elemento seleccionado en una interfaz separada.■ Se genera un modelo 3D del elemento seleccionado en el área de trabajo.
Situaciones de error	No aplica.
Actores	Usuario
Disparadores	El usuario toca un elemento químico en la tabla periódica interactiva utilizando el hand tracking.
Proceso estándar	<ol style="list-style-type: none">1. El usuario toca un elemento químico en la tabla periódica interactiva.2. El sistema detecta la selección y resalta visualmente el elemento.3. El sistema recupera información detallada sobre el elemento seleccionado (nombre, símbolo, número atómico, masa atómica, etc.) y la muestra en una interfaz de usuario separada.4. El sistema genera un modelo 3D del elemento seleccionado y lo coloca en el área de trabajo.

Continúa en la siguiente página

Caso de Uso	Seleccionar Elemento Químico
--------------------	------------------------------

Procesos alternativos	No aplica.
------------------------------	------------

Tabla 3.10: Descripción del caso de uso Seleccionar Elemento Químico

Caso de Uso	Crear Compuesto
--------------------	-----------------

Descripción corta	El usuario crea un compuesto químico combinando elementos seleccionados de la tabla periódica.
--------------------------	--

Precondición	<ul style="list-style-type: none">■ El usuario ha seleccionado los elementos químicos necesarios para formar el compuesto.■ Los elementos seleccionados están presentes en el área de trabajo.
---------------------	---

Postcondición	<ul style="list-style-type: none">■ El sistema genera un modelo 3D del compuesto formado.■ El modelo 3D del compuesto se coloca en el área de trabajo.■ Los modelos 3D de los elementos utilizados se eliminan del área de trabajo.
----------------------	---

Situaciones de error	<ul style="list-style-type: none">■ Combinación Inválida: El usuario intenta combinar elementos que no forman un compuesto válido.■ Elementos Insuficientes: El usuario no ha seleccionado todos los elementos necesarios para formar el compuesto.
-----------------------------	--

Continúa en la siguiente página

Caso de Uso	Crear Compuesto
Actores	Usuario
Disparadores	El usuario toca un elemento químico en la tabla periódica interactiva utilizando el hand tracking.
Proceso estándar	<ol style="list-style-type: none"> 1. El usuario indica al sistema que desea crear un compuesto. 2. El sistema verifica si la combinación de elementos es válida y si todos los elementos necesarios están presentes. 3. Si la combinación es válida, el sistema genera un modelo 3D del compuesto y lo coloca en el área de trabajo. 4. El sistema elimina los modelos 3D de los elementos utilizados.
Procesos alternativos	<p>2A. Combinación Inválida:</p> <ul style="list-style-type: none"> ■ El sistema detecta que la combinación de elementos no es válida. ■ El sistema muestra un mensaje de error indicando que la combinación no es posible. ■ El sistema permite al usuario seleccionar otros elementos o cancelar la creación del compuesto. <p>2B. Elementos Insuficientes:</p> <ul style="list-style-type: none"> ■ El sistema detecta que faltan elementos para formar el compuesto. ■ El sistema muestra un mensaje de error indicando los elementos faltantes. ■ El sistema permite al usuario seleccionar los elementos faltantes o cancelar la creación del compuesto.

Tabla 3.11: Descripción del caso de uso Crear Compuesto

Caso de Uso	Calentar Compuesto
--------------------	--------------------

Descripción corta	El usuario calienta un compuesto utilizando un mechero Bunsen.
--------------------------	--

Precondición	<ul style="list-style-type: none">■ El mechero Bunsen está disponible y encendido.■ El usuario se encuentra en una etapa del experimento que requiere calentar un compuesto.
---------------------	---

Postcondición	El compuesto válido se calienta y se muestran los cambios visuales y auditivos correspondientes.
----------------------	--

Situaciones de error	El usuario intenta calentar una sustancia o compuesto no válido.
-----------------------------	--

Actores	Usuario
----------------	---------

Disparadores	El usuario acerca el compuesto al mechero Bunsen.
---------------------	---

Proceso estándar	<ol style="list-style-type: none">1. El usuario selecciona el mechero Bunsen y lo acerca a la sustancia o compuesto (Controlador).2. El sistema verifica si la sustancia o compuesto es válido para calentar (Modelo).3. Si es válido, el sistema detecta la proximidad y activa el calentamiento (Modelo).4. El sistema simula el efecto del calor en el compuesto, mostrando cambios visuales (burbujas, cambio de color) y auditivos (sonido de ebullición) (Vista).
-------------------------	--

Continúa en la siguiente página

Caso de Uso	Calentar Compuesto
--------------------	--------------------

Procesos alternativos	2A. Sustancia o Compuesto No Válido: Si la sustancia o compuesto no es válido, el sistema muestra un mensaje de error.
------------------------------	---

Tabla 3.12: Descripción del caso de uso Calentar Compuesto

Caso de Uso	Enfriar Compuesto
Descripción corta	El usuario sumerge un vaso de precipitados que contiene una solución en un baño de agua para disminuir su temperatura y observar los cambios resultantes.
Precondición	<ul style="list-style-type: none"> ■ El vaso de precipitados con la solución a enfriar está presente en el área de trabajo. ■ El baño de agua está disponible. ■ El usuario se encuentra en una etapa del experimento que requiere enfriar la solución
Postcondición	El compuesto válido se enfriá y se muestran los cambios visuales correspondientes.
Situaciones de error	El usuario intenta enfriar una sustancia o compuesto no válido.
Actores	Usuario
Disparadores	El usuario acerca el recipiente al baño de agua.

Caso de Uso

Enfriar Compuesto

Proceso estándar

1. El usuario selecciona el recipiente que contiene el compuesto y lo sumerge en el baño de hielo (Controlador).
2. El sistema verifica si la sustancia o compuesto es válido para enfriar (Modelo).
3. Si es válido, el sistema detecta la inmersión y activa el enfriamiento (Modelo).
4. El sistema simula el efecto del enfriamiento en el compuesto (Vista).

Procesos alternativos

- 2A. **Sustancia o Compuesto No Válido:** Si la sustancia o compuesto no es válido, el sistema muestra un mensaje de error.

Tabla 3.13: Descripción del caso de uso Enfriar Compuesto

Caso de Uso

Combinar Sustancias

Descripción corta

El usuario combina dos o más sustancias en un nuevo recipiente para iniciar una reacción química.

Precondición

Las sustancias a combinar están disponibles en el área de trabajo.

Postcondición

Las sustancias válidas se combinan en un nuevo recipiente y se produce la reacción química, mostrando los cambios visuales y auditivos correspondientes.

Situaciones de error

El usuario intenta combinar sustancias no válidas.

Continúa en la siguiente página

Caso de Uso	Combinar Sustancias
Actores	Usuario
Disparadores	El usuario arrastra las sustancias seleccionadas a un nuevo recipiente.
Proceso estándar	<ol style="list-style-type: none"> 1. El usuario selecciona las sustancias que desea combinar y las arrastra al recipiente (Controlador). 2. El sistema verifica la validez de las sustancias para ser combinadas (Modelo). 3. Si son compatibles, el sistema simula la combinación y la reacción química, mostrando cambios visuales y auditivos (Vista).
Procesos alternativos	<p>2A. Combinación Inválida: Si las sustancias no son compatibles, el sistema muestra un mensaje de error.</p>

Tabla 3.14: Descripción del caso de uso Combinar Sustancias

Caso de Uso	Desechar elemento
Descripción corta	El usuario elimina un elemento del área de trabajo.
Precondición	<ul style="list-style-type: none"> ■ Existencia del Elemento: El elemento a desechar debe estar presente y visible en el área de trabajo. ■ Área de Desecho Habilitada: El área de desecho debe estar habilitada y accesible para el usuario.

Continúa en la siguiente página

Caso de Uso	Desechar elemento
Postcondición	El elemento o compuesto se elimina del área de trabajo.
Situaciones de error	El usuario intenta desechar un elemento que el sistema no permite eliminar en ese momento del experimento.
Actores	Usuario
Disparadores	El usuario deposita el elemento en el área de desecho.
Proceso estándar	<ol style="list-style-type: none"> 1. Inicio: El usuario toma el elemento o compuesto a desechar. 2. Movimiento: El usuario mueve el elemento o compuesto hacia el área de desecho. 3. Soltar: El usuario suelta el elemento o compuesto dentro de los límites del área de desecho. 4. Validación: El sistema valida que el elemento o compuesto puede ser desecharado en ese momento del experimento. 5. Eliminación: El sistema elimina el elemento o compuesto del área de trabajo y de cualquier estructura de datos asociada.
Procesos alternativos	<ol style="list-style-type: none"> 4A. Validación: El sistema detecta que el elemento o compuesto no puede ser desecharado en ese momento. 4B. Notificación: El sistema muestra un mensaje de error claro indicando que el elemento o compuesto no puede ser desecharado en ese momento. 4C. Cancelación: El elemento o compuesto regresa a su posición original en el área de trabajo.

Tabla 3.15: Descripción del caso de uso Desechar elemento

Caso de Uso	Experimentación y Reacción
Descripción corta	El usuario interactúa con los compuestos y el instrumental para realizar la experimentación y reacción química, mientras el sistema simula los efectos.
Precondición	<ul style="list-style-type: none">■ Compuestos disponibles: Los compuestos químicos necesarios para el experimento se seleccionaron y están en el área de trabajo virtual.■ Instrumental disponible: El instrumental virtual requerido (matraces, pipetas, mecheros, etc.) está presente y accesible en el área de trabajo.
Postcondición	<ul style="list-style-type: none">■ Resultados simulados: El sistema ha simulado los resultados de la reacción o experimento químico realizado por el usuario.■ Visualización de resultados: El usuario puede observar los cambios en los compuestos, la formación de nuevos productos, cambios de temperatura, etc., en la interfaz del sistema.
Situaciones de error	<ul style="list-style-type: none">■ Compuestos Insuficientes: El usuario intenta realizar un experimento sin todos los compuestos necesarios.■ Acciones Inválidas: Si el usuario intenta realizar una acción no permitida por el diseño del experimento, el sistema lo notificará y detendrá la simulación.
Actores	Usuario
Disparadores	El usuario comienza a interactuar con los compuestos o el instrumental virtual.

Continúa en la siguiente página

Caso de Uso	Experimentación y Reacción
--------------------	----------------------------

Proceso estándar

1. **Inicio:** El usuario toma el elemento o compuesto a desechar.
 2. **Movimiento:** El usuario mueve el elemento o compuesto hacia el área de desecho.
 3. **Soltar:** El usuario suelta el elemento o compuesto dentro de los límites del área de desecho.
 4. **Validación:** El sistema valida que el elemento o compuesto puede ser desechar en ese momento del experimento.
 5. **Eliminación:** El sistema elimina el elemento o compuesto del área de trabajo y de cualquier estructura de datos asociada.
-

Procesos alternativos

- 4A. Validación: El sistema detecta que el elemento o compuesto no puede ser desechar en ese momento.
 - 4B. Notificación: El sistema muestra un mensaje de error claro indicando que el elemento o compuesto no puede ser desechar en ese momento.
 - 4C. Cancelación: El elemento o compuesto regresa a su posición original en el área de trabajo.
-

Tabla 3.16: Descripción del caso de uso Experimentación y Reacción

Caso de Uso	Main Menu
--------------------	-----------

Descripción corta

El usuario sale del experimento actual y regresa al menú de bienvenida.

Precondición

El usuario se encuentra en el Menú del escenario de un experimento.

Continúa en la siguiente página

Caso de Uso	Main Menu
Postcondición	El experimento actual se cierra y el usuario regresa al menú de bienvenida.
Situaciones de error	Ninguna
Actores	Usuario
Disparadores	El usuario selecciona la opción "Main Menu." en el menú.
Proceso estándar	<ol style="list-style-type: none"> 1. El sistema detecta la selección de la opción "Main Menu". 2. El sistema detiene la simulación del experimento actual. 3. El sistema regresa al menú de bienvenida.
Procesos alternativos	Ninguno.

Tabla 3.17: Descripción del caso de uso Main Menu

Caso de Uso	Colocar instrumental en soporte
Descripción corta	El usuario coloca un instrumento de laboratorio en un soporte designado.
Precondición	El usuario está sosteniendo un instrumento de laboratorio con la mano y hay un soporte disponible.

Continúa en la siguiente página

Caso de Uso	Colocar instrumental en soporte
Postcondición	El instrumento de laboratorio se coloca en el soporte y se libera de la mano del usuario.
Situaciones de error	No hay un soporte disponible o el usuario no está sosteniendo un instrumento compatible con el soporte.
Actores	Usuario
Disparadores	El usuario acerca el instrumento al soporte y realiza el gesto de "soltar".
Proceso estándar	<ol style="list-style-type: none"> 1. El sistema detecta que el usuario está acercando un instrumento a un soporte. 2. El sistema verifica si el soporte está disponible y si el instrumento es compatible. 3. Si el soporte está disponible y el instrumento es compatible, el sistema coloca el instrumento en el soporte y lo libera de la mano del usuario.
Procesos alternativos	<p>3A. Si el soporte no está disponible o el instrumento no es compatible, el sistema muestra un mensaje de error al usuario..</p>

Tabla 3.18: Descripción del caso de uso Colocar instrumental en soporte

3.5.2. Diagrama de Clases

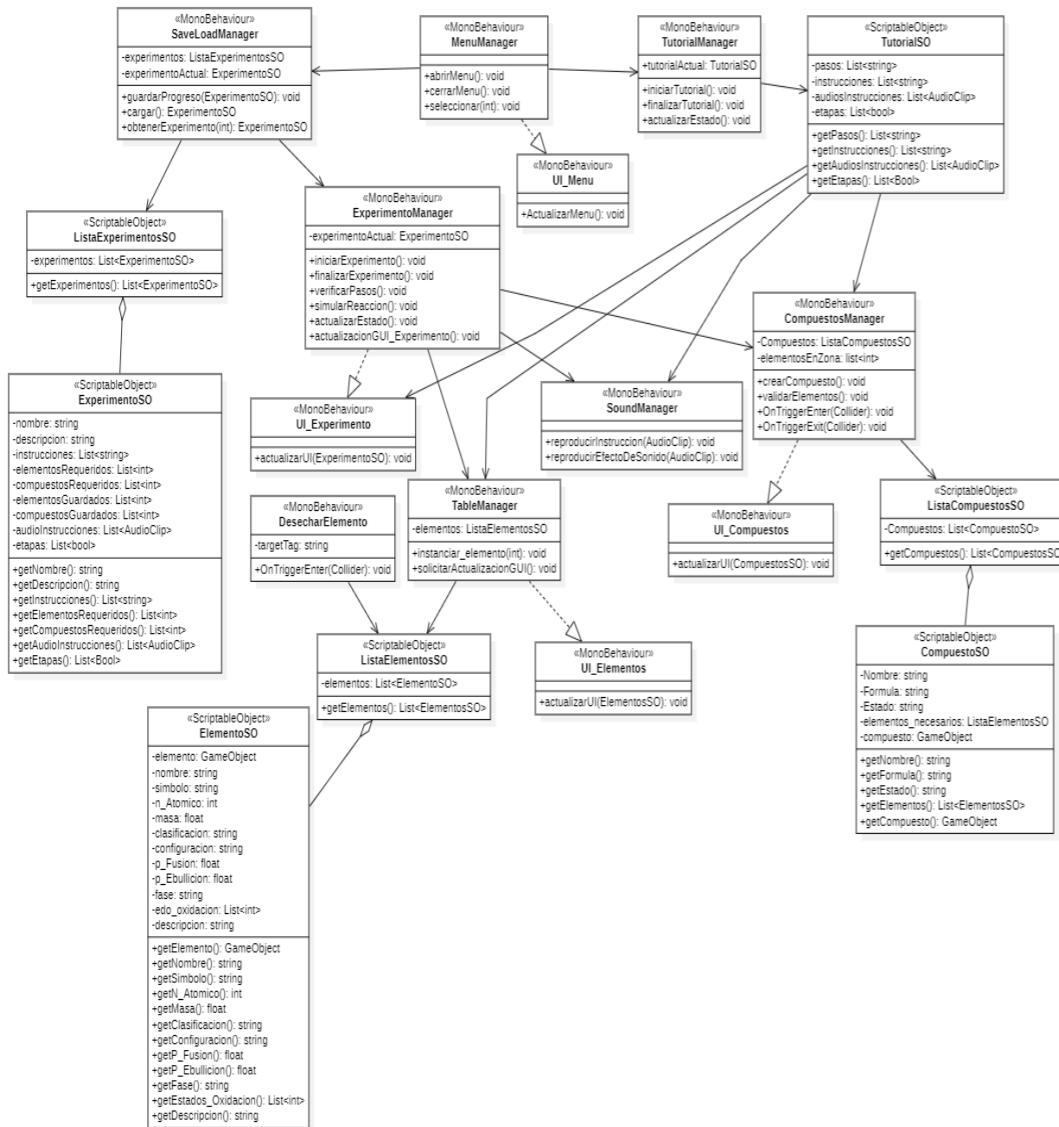


Figura 3.14: Diagrama de Clases

CAPÍTULO 4

Desarrollo

4.1. Estructura del Simulador

En el desarrollo del prototipo de simulador de laboratorio de Química Inorgánica en Realidad Virtual, la organización del proyecto en Unity desempeña un papel crucial para mantener un entorno de desarrollo estructurado, escalable y mantenable. Al crear un proyecto nuevo en Unity, se genera una carpeta principal llamada Assets, que contiene todos los recursos necesarios para el correcto funcionamiento del simulador.

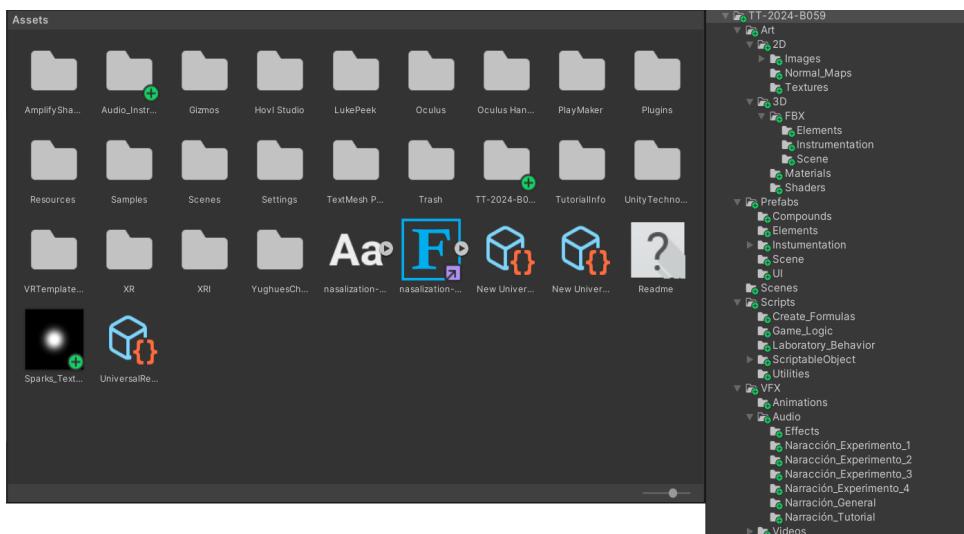


Figura 4.1: Assets del Simulador

Esta carpeta se subdivide en varias carpetas específicas según su propósito, como se detalla a continuación:

I. Art:

Esta carpeta contiene los recursos gráficos utilizados en el simulador, divididos en dos subcarpetas principales:

- **2D:** Incluye imágenes, mapas normales y texturas utilizadas para enriquecer los elementos visuales del entorno.

- **3D:** Contiene modelos tridimensionales organizados en:
 - FBX: Modelos de elementos, instrumental y escenas del laboratorio.
 - Materials: Materiales aplicados a los modelos para simular texturas y propiedades físicas.
 - Shaders_Graphs: Gráficos de sombreado personalizados que mejoran el realismo del entorno virtual.

II. *Prefabs:*

Los prefabs son objetos prediseñados con configuraciones preestablecidas, lo que permite reutilizarlos en múltiples escenas. Están organizados en:

- **Compounds:** Compuestos químicos modelados y configurados con comportamientos específicos.
- **Elements:** Elementos químicos individuales listos para ser utilizados en las simulaciones.
- **Instrumentation:** Herramientas e instrumentos del laboratorio como matraces, tubos de ensayo y mecheros.
- **Scene:** Componentes específicos para gestionar aspectos visuales y funcionales de las escenas.
- **UI:** Elementos de interfaz de usuario diseñados para facilitar la interacción.

III. *Scenes:*

Esta carpeta contiene todas las escenas creadas para el simulador

- **Experimentos:** Cada experimento tiene su propia escena, diseñada específicamente para simular procedimientos y reacciones químicas.
- **Tutorial:** Provee instrucciones guiadas para familiarizar al usuario con el simulador y las actividades.
- **Hub:** Actúa como un menú principal donde el usuario puede navegar hacia diferentes experimentos o tutoriales.

IV. *Scripts:*

Contiene todos los scripts de código fuente en lenguaje C#, organizados según su funcionalidad en las siguientes subcarpetas:

- **Create Formulas:** Scripts para crear y gestionar las fórmulas químicas utilizadas en el simulador.
- **Game Logic:** Lógica general del simulador, como eventos y reglas específicas.
- **Laboratory Behavior:** Comportamientos de los elementos interactivos del laboratorio.
- **ScriptableObject:** Configuraciones y datos reutilizables en formato ScriptableObject.
- **Utilities:** Scripts auxiliares que apoyan funcionalidades generales del simulador.

Los scripts se desarrollan siguiendo principios de programación orientada a objetos, aplicando el uso de máquinas de estado para la gestión de escenas y experimentos.

V. VFX:

Esta carpeta contiene los efectos visuales y auditivos que mejoran la experiencia inmersiva del usuario:

- **Animations:** Animaciones aplicadas a modelos y elementos interactivos.
- **Audio:** Archivos de sonido organizados en:
 - Effects: Efectos de sonido específicos.
 - Narraciones: Audio dividido en narraciones para tutoriales y experimentos.
- **Videos:** Clips de video utilizados para complementar la experiencia educativa.

4.2. Ambiente Virtual

El ambiente virtual desarrollado para el simulador proporciona un espacio interactivo e inmersivo donde los usuarios pueden realizar diversos experimentos de química inorgánica. Este entorno simula un laboratorio moderno e incluye los elementos necesarios para recrear prácticas científicas de manera eficiente y educativa.

4.2.1. Diseño del entorno 3D

El laboratorio consta de los siguientes elementos principales:

- **Mesa de trabajo:** Área central donde se llevan a cabo los experimentos. Diseñada para proporcionar un espacio virtual amplio y funcional que facilita la manipulación de los objetos.
- **Modelos atómicos de los elementos:** Representaciones tridimensionales de los átomos, diseñadas para ilustrar sus estructuras y propiedades.

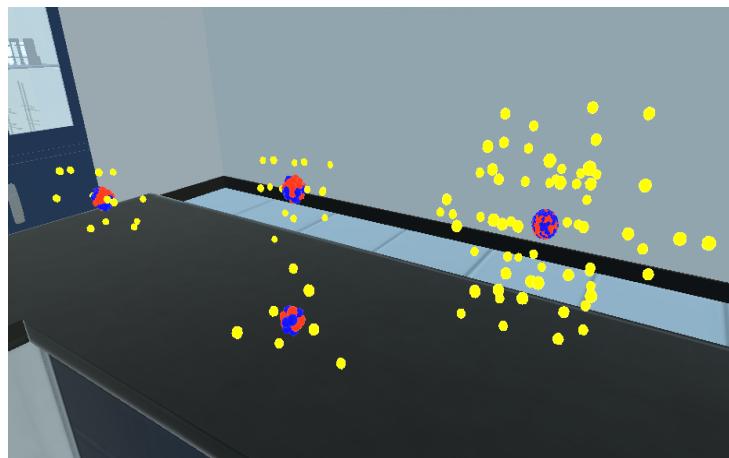


Figura 4.2: Modelos de Elementos Químicos

- **Tabla periódica interactiva:** Permite al usuario explorar los elementos químicos de forma dinámica. Al seleccionar un elemento, se despliega información relevante y un modelo atómico tridimensional.

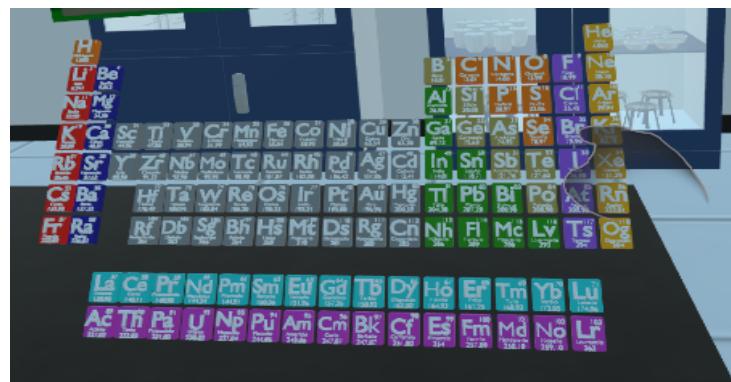


Figura 4.3: Tabla Periódica

- **Instrumental necesario:** Incluye matraces, tubos de ensayo, mecheros y demás herramientas esenciales, modeladas en detalle para reflejar con precisión el equipo utilizado en laboratorios reales.



Figura 4.4: Instrumental de Laboratorio

- **Modelos de compuestos químicos:** Reproducciones de los compuestos que se generan durante los experimentos, mostrando sus características visuales de manera precisa.

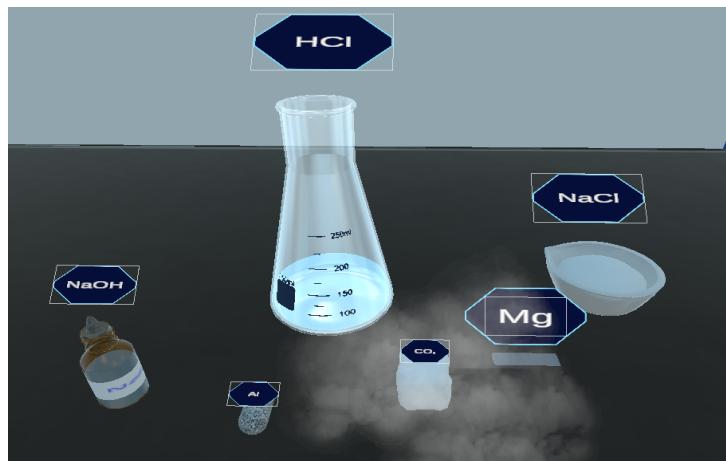


Figura 4.5: Compuestos Químicos

Todo el modelado 3D fue desarrollado utilizando Blender, una herramienta ampliamente reconocida en la industria por su capacidad para crear modelos tridimensionales detallados y eficientes. Las texturas y materiales de estos modelos se trabajaron en GIMP.

4.2.2. Efectos Visuales (VFX)

Para mejorar la inmersión y reforzar la representación de las reacciones químicas y los fenómenos experimentales, se diseñaron una variedad de efectos de partículas. Entre los principales se encuentran:

- **Humo y Vapores:** Representan las liberaciones de gases en ciertas reacciones químicas.
- **Fuego y Chispas:** Simulan encendidos de mecheros y otras reacciones exotérmicas.
- **Niebla:** Utilizada como un efecto visual en reacciones específicas.

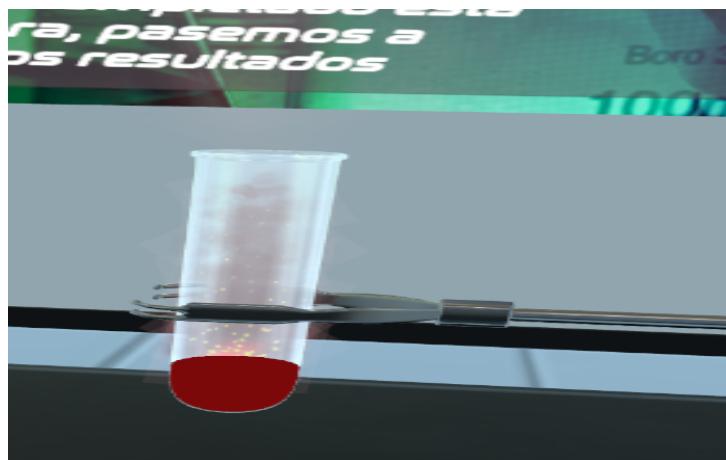


Figura 4.6: Efectos de Partículas

Estos efectos se crearon utilizando el Unity Particle System y shaders personalizados para lograr una integración fluida con el entorno.

4.2.3. Efectos de Sonido

El sonido juega un papel fundamental en la inmersión sensorial del simulador. Se incorporaron diversos efectos sonoros para acompañar las interacciones y eventos del laboratorio, incluyendo:

- **Cristales:** Simulan el sonido de recipientes de vidrio al ser manipulados o chocados.
- **Burbujeos y flujo de agua:** Representan procesos químicos y líquidos en movimiento.
- **Fuego y gas:** Emulan encendidos de mecheros y escapes controlados de gases.

Los sonidos fueron seleccionados y editados para sincronizarse con precisión en tiempo real mediante el sistema de eventos de Unity. Esto asegura que cada acción del usuario esté acompañada de una respuesta auditiva coherente y realista.

4.2.4. Instrucciones guiadas con voz

Para las instrucciones paso a paso, se utilizó el servicio en línea SpeechGen.io, el cual permitió generar voces claras y dinámicas en varios idiomas. Este sistema se integró para:

- Proporcionar guías auditivas que refuerzen las instrucciones visuales.
- Sincronizar las instrucciones de voz con las acciones y eventos dentro del laboratorio, mejorando la accesibilidad y la experiencia del usuario.

4.3. Diseño de Interfaz de Usuario

El diseño de la interfaz de usuario (IU) para el simulador se desarrolló con un enfoque en maximizar la funcionalidad y la accesibilidad dentro del entorno de realidad virtual (RV), adaptándose a las necesidades de un laboratorio químico interactivo. Estas interfaces, aunque densas en información, fueron optimizadas para que los usuarios puedan acceder fácilmente a datos relevantes y operar de manera eficiente en un entorno inmersivo.

4.3.1. Hub de selección

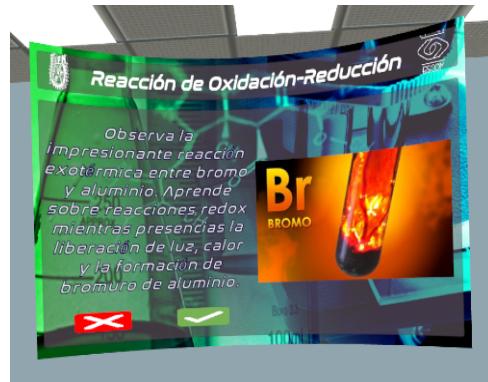
Sirve como el punto de entrada al simulador, diseñado para ofrecer una navegación jerarquía y estructurada:

- **Pantalla inicial:** Contiene dos botones principales: Tutorial y Experimentos.
- **Pantalla de selección de experimentos:** Presenta 4 botones, cada uno correspondiente a un experimento disponible y un quinto botón para regresar al menú principal.
- **Pantalla Tutorial-Experimentos:** Proporciona una descripción general de la selección, con su imagen de referencia y dos opciones de navegación, regresar y confirmar.



(a) Pantalla Principal Hub

(b) Pantalla Selección Experimento



(c) Pantalla Información Experimento

Figura 4.7: IU Hub

4.3.2. Información principal

Centraliza la información operativa de cada experimento, proporcionando:

- Instrucciones detalladas
- Soporte visual
- Ecuaciones químicas

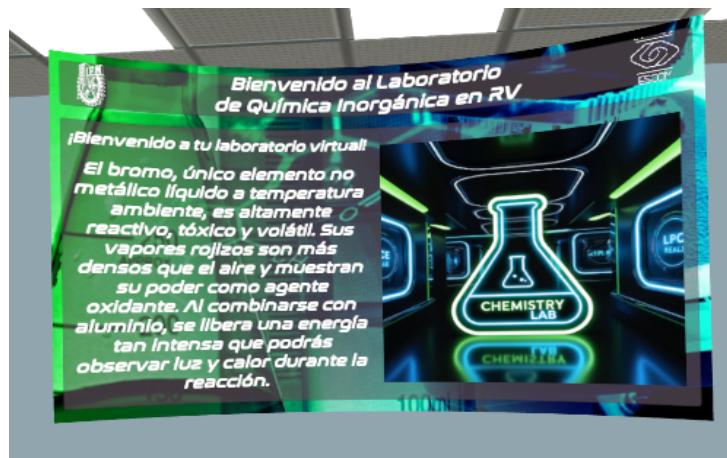


Figura 4.8: IU Principal

4.3.3. Elementos

Se focaliza en proporcionar datos contextuales sobre el último elemento químico seleccionado



Figura 4.9: IU Elementos

4.3.4. Compuestos

Presenta información sobre el último compuesto generado.



Figura 4.10: IU Compuestos

4.3.5. Área de creación

Gestiona la combinación de elementos químicos.

- **Indicador de contenido:** Lista dinámica de los elementos colocados en el área de creación.
- **Validación automática:** Notificación visual que indica si la combinación es válida para generar un compuesto
- **Botón Crear:** Activo únicamente cuando los elementos combinados forman un compuesto válido. Al presionarlo, se genera el compuesto y se actualiza la interfaz.



Figura 4.11: IU Creación

4.3.6. Teclado numérico interactivo

Se sincroniza con las ecuaciones químicas mostradas en la IU de información principal, permitiendo la edición en tiempo real.



Figura 4.12: IU Numpad

4.3.7. Pantalla final

Marca la conclusión de un experimento e incluye dos acciones clave:

- Repetir experimento: Reinicia el procedimiento desde el principio.
- Regresar al hub inicial: Permite al usuario explorar otras opciones o finalizar la sesión.



Figura 4.13: IU Pantalla Final

4.4. Interacción y Gestos

La interacción dentro del simulador se diseñó para aprovechar las capacidades del seguimiento de manos (hand tracking) ofrecidas por el SDK de Oculus. Esto permite una experiencia intuitiva y natural para el usuario, integrando gestos básicos que emulan acciones físicas reales y simplifican la navegación en el entorno virtual.

4.4.1. Interacción con el sistema

Gracias a las herramientas proporcionadas por el SDK de Oculus, los objetos interactivos fueron configurados con los scripts Grabbable y Hand Grab Interactable. Esto habilita dos formas principales de interacción con los objetos en el entorno:

- Sujeción completa: El usuario puede interactuar con los objetos cerrando completamente su mano alrededor de ellos, replicando la acción física de sujetar un objeto.



Figura 4.14: Sujeción completa

- Gesto de pinchar: El usuario junta el dedo índice y el pulgar (gesto de pinchar) para manipular objetos más pequeños o realizar interacciones precisas, como seleccionar botones o modificar valores.



Figura 4.15: Gesto de pinchar

Este gesto específico también se utiliza para interactuar con coeficientes en las ecuaciones químicas y en la selección de botones mediante un apuntador virtual. Esto asegura que las interacciones sean ergonómicas y precisas, incluso en actividades detalladas.

4.4.2. Gestos específicos

Se implementaron gestos adicionales para optimizar la interacción con el laboratorio:

- Selección directa con el dedo índice: Al tocar directamente la tabla periódica interactiva o botones cercanos, el sistema reconoce el contacto del dedo índice y activa la acción correspondiente. Esto elimina la necesidad de punteros adicionales para elementos de acceso rápido.



Figura 4.16: Interacción con botones cercanos

- Centrado del escenario: Se habilitó un gesto general estándar de Oculus para centrar el entorno a la vista del usuario. Este gesto garantiza que el laboratorio virtual se ajuste correctamente al campo de visión, mejorando la comodidad y la orientación espacial del usuario.

El uso de gestos básicos asegura que los usuarios puedan interactuar con el entorno sin requerir controladores físicos. Esto reduce la curva de aprendizaje y mejora la experiencia inmersiva.



Figura 4.17: Gesto para centrar el entorno

4.5. Integración del SDK de Oculus

El desarrollo del simulador requirió la implementación del SDK de Oculus, una herramienta clave para habilitar capacidades de realidad virtual inmersiva. Este SDK proporciona un modelo completo que integra cámaras virtuales, controladores y seguimiento de manos (hand tracking) en un entorno preconfigurado.

Para este proyecto, se utilizó el módulo de hand tracking como base principal de interacción, aprovechando su capacidad para detectar y procesar movimientos naturales de las manos sin necesidad de controladores físicos. Este enfoque es particularmente relevante para la interacción con objetos y elementos virtuales descritos en la Sección 4.2: Ambiente Virtual y la Sección 4.4: Interacción y Gestos.

La integración del SDK de Oculus simplificó significativamente el desarrollo del simulador, proporcionando una base sólida para implementar las interacciones y funcionalidades principales sin necesidad de construir estas capacidades desde cero.

CAPÍTULO 5

Implementación

En este capítulo se describen las principales escenas que componen el simulador, las máquinas de estado que gestionan el flujo de cada una, las acciones que los usuarios pueden realizar, y los scripts más relevantes que soportan la funcionalidad del sistema. El simulador está diseñado para guiar al usuario a través de una experiencia educativa interactiva y estructurada, permitiendo un aprendizaje inmersivo en el contexto de un laboratorio de química inorgánica.

5.1. Fases de los Experimentos

Los experimentos en el simulador están estructurados en cinco fases principales, cada una diseñada para facilitar la comprensión y ejecución de los procesos químicos involucrados. Estas fases aseguran que el usuario avance de manera lógica y progresiva, consolidando conocimientos y habilidades a lo largo de la actividad.

5.1.1. Introducción

En esta fase inicial, se da la bienvenida al usuario con una explicación clara del experimento y sus objetivos principales. Se destaca el contexto químico de la actividad, los materiales necesarios, y una descripción breve pero detallada del fenómeno que se observará. Esta etapa motiva al usuario y establece las bases para comprender los conceptos químicos que serán explorados durante el experimento.

5.1.2. Balanceo de Ecuaciones

El usuario aprende sobre las reacciones químicas involucradas en el experimento a través del análisis y balanceo de ecuaciones. Se le proporciona una ecuación inicial que deberá balancear para garantizar que la cantidad de átomos en ambos lados sea equivalente. El sistema guía al usuario paso a paso, explicando los conceptos básicos y proporciones estequiométricas, mientras utiliza el teclado numérico interactivo para ingresar los coeficientes correctos.

5.1.3. Selección y Creación

El usuario selecciona los elementos necesarios desde la tabla periódica virtual y los combina en el área de creación para formar los compuestos requeridos. A través de gestos intuitivos, como el de pinchar o agarrar, se seleccionan materiales como líquidos o sólidos, que luego se transforman en objetos interactivos listos para el experimento. El sistema proporciona retroalimentación visual para indicar que los compuestos se han preparado correctamente antes de proceder a la experimentación.

5.1.4. Experimentación

En esta etapa, el usuario lleva a cabo el experimento siguiendo instrucciones precisas y detalladas. Se realizan acciones como agregar líquidos o sólidos a recipientes, encender un mechero, calentar materiales, o combinar reactivos, observando los cambios físicos y químicos en tiempo real. Cada acción está respaldada por efectos visuales y sonoros que simulan fenómenos como la liberación de gases, formación de humo, cambios de color y producción de calor o luz, creando una experiencia inmersiva y educativa.

5.1.5. Explicación

Finalmente, el usuario recibe una explicación detallada de los resultados obtenidos durante el experimento. Esta fase resalta los principios químicos involucrados, como reacciones redox, formación de compuestos o cambios energéticos. Se utilizan gráficos y representaciones visuales para reforzar el aprendizaje, asegurando que el usuario comprenda plenamente el fenómeno observado y su relevancia en el contexto de la química.

5.2. Escenas Principales

5.2.1. Hub

En la escena del Hub, se implementó una máquina de estados finita (FSM) para gestionar las transiciones entre las diferentes opciones disponibles, como el acceso al tutorial o a los experimentos. Este diseño permite una navegación fluida y estructurada, asegurando que el usuario pueda seleccionar y avanzar hacia las actividades deseadas de manera eficiente.

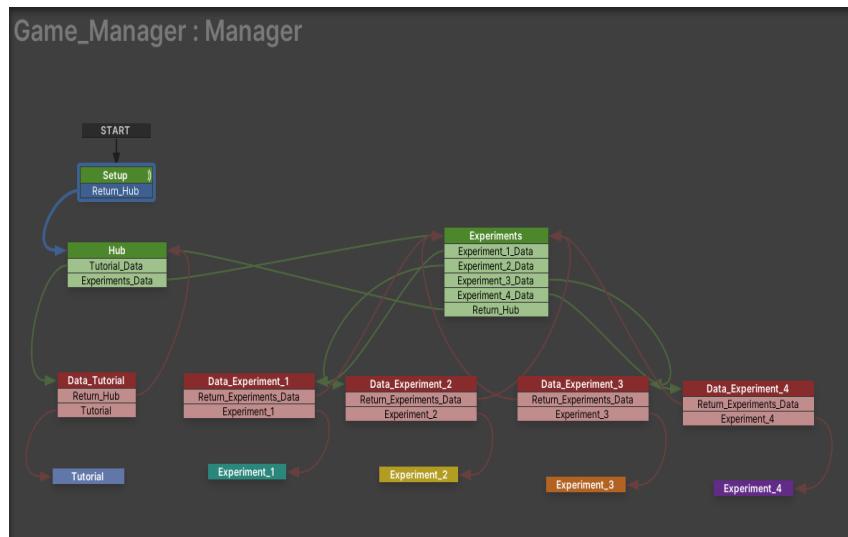


Figura 5.1: Flujo de la escena Hub

Para implementar y manejar esta máquina de estados, se utilizó el script `MenuController`, que se encuentra detallado en la (Sección B.1). Este script es responsable de la lógica principal, como el manejo de eventos, la actualización de datos en la interfaz y la ejecución de las transiciones correspondientes dentro del flujo definido por la FSM. La Figura 5.1 muestra el diagrama de la máquina de estados que estructura esta escena.

5.2.2. Tutorial

La escena de Tutorial utiliza una FSM para organizar las etapas de aprendizaje, desde la introducción hasta la realización de interacciones clave. Esta máquina guía al usuario a través de un proceso secuencial diseñado para familiarizarlo con las mecánicas del simulador y las herramientas virtuales.

La escena comienza con una bienvenida que introduce al usuario a las herramientas y conceptos básicos del simulador. Luego, se presentan tareas prácticas que incluyen el balanceo de ecuaciones, la selección y mezcla de sustancias, y el encendido y calentamiento de materiales, todo respaldado por retroalimentación visual y auditiva. Finalmente, la escena concluye con una despedida que refuerza los conceptos aprendidos y permite al usuario volver al menú principal.

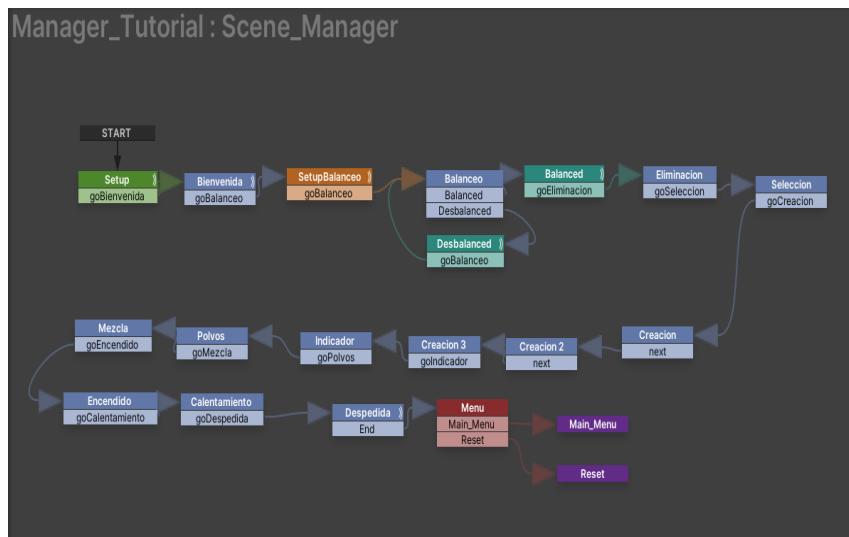


Figura 5.2: Flujo de la escena Tutorial

5.2.3. Experimento 1

La escena del Experimento 1 está diseñada para gestionar tareas específicas relacionadas con el balanceo de ecuaciones químicas y la manipulación de sustancias. La FSM controla cada etapa del experimento, permitiendo avanzar solo cuando las condiciones necesarias se cumplen.

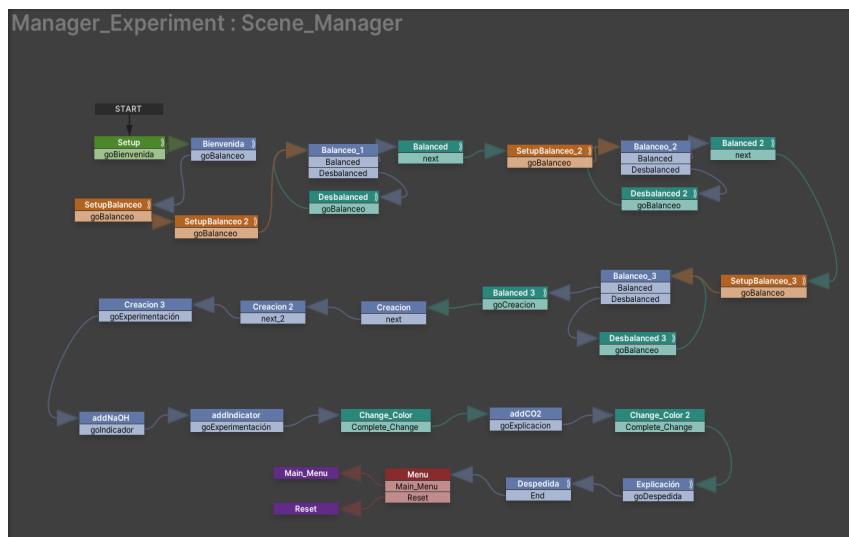


Figura 5.3: Flujo de la escena Experimento 1

El flujo del experimento incluye pasos como el balanceo de ecuaciones químicas, la selección de sustancias específicas y su combinación, y la observación de cambios visuales en los materiales. Cada etapa está respaldada por retroalimentación visual y auditiva para mejorar la comprensión del usuario. Además, se utilizan efectos visuales, como cambios de color, para simular reacciones químicas.

El script `Experiment_1_Manager`, documentado en la Sección B.2 valida las acciones del usuario, como la correcta combinación de sustancias, y activa eventos en la FSM para reflejar el progreso en tiempo real. Además, se manejan efectos visuales para representar las reacciones químicas de forma precisa. La estructura del flujo de esta escena se puede observar en la Figura 5.3.

5.2.4. Experimento 2

En el Experimento 2, la FSM organiza actividades que involucran la validación de mezclas químicas y la manipulación de temperatura. Los estados aseguran que las condiciones del experimento, como el tiempo de calentamiento o las combinaciones correctas, se cumplan antes de proceder.

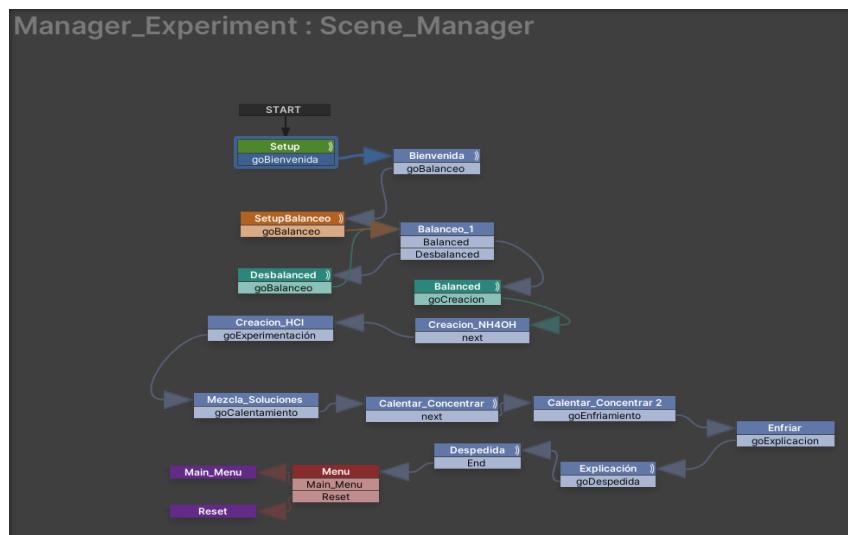


Figura 5.4: Flujo de la escena Experimento 2

El script `Experiment_2_Manager`, documentado en la Sección B.3, implementa la lógica para monitorear estos parámetros y activar transiciones en la FSM, además de generar efectos visuales y auditivos que simulan las reacciones químicas.

5.2.5. Experimento 3

La FSM del Experimento 3 gestiona las combinaciones de sustancias y la activación de reacciones exotérmicas, estructurando el flujo del experimento para garantizar que cada acción del usuario sea validada antes de avanzar.

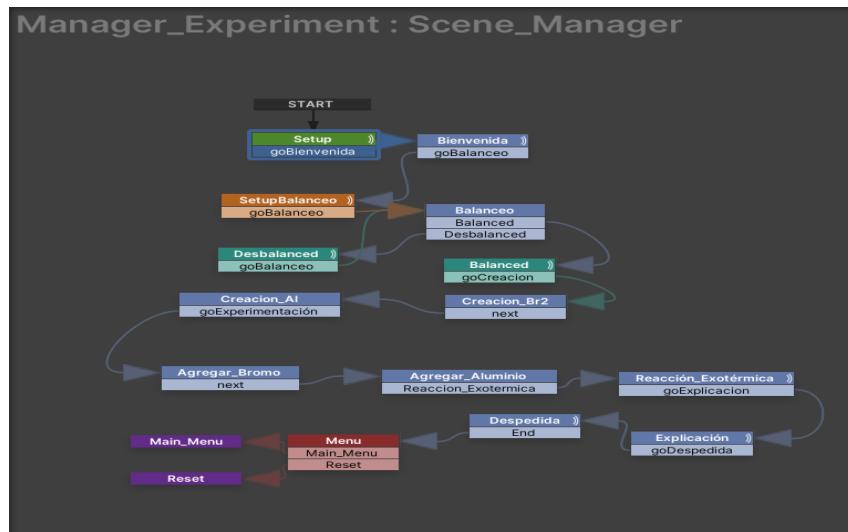


Figura 5.5: Flujo de la escena Experimento 3

El script Experiment_3_Manager, documentado en la Sección B.4, se encarga de verificar las combinaciones realizadas y de controlar eventos visuales, como el llenado de recipientes, ofreciendo una experiencia interactiva y alineada con los objetivos del experimento.

5.2.6. Experimento 4

El Experimento 4 utiliza una FSM para gestionar tareas relacionadas con la manipulación de magnesio y dióxido de carbono sólido (hielo seco). Las etapas incluyen la preparación inicial, el encendido del magnesio, y la observación de una reacción exotérmica que genera luz intensa y subproductos químicos.

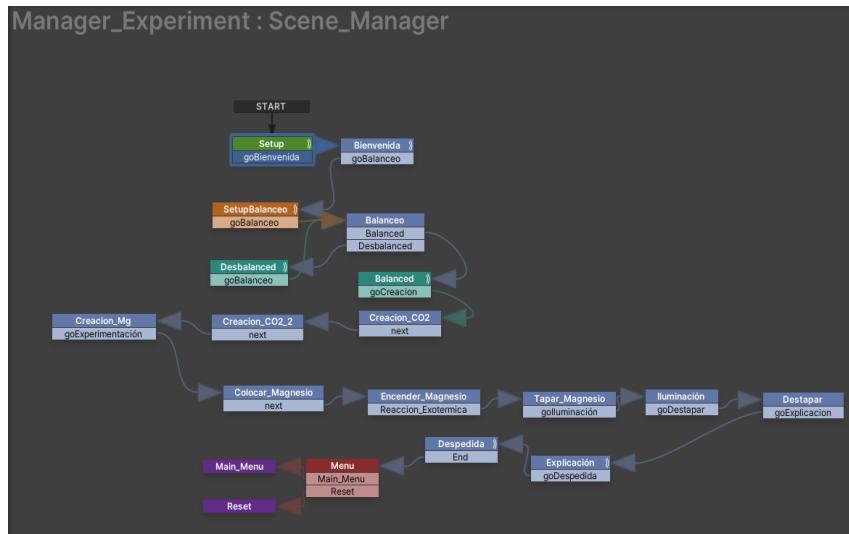


Figura 5.6: Flujo de la escena Experimento 4

El script Experiment_4_Manager, documentado en la Sección B.5, valida las acciones del usuario, como la correcta combinación de sustancias, y coordina eventos visuales y auditivos que simulan los procesos químicos involucrados. Este enfoque asegura que la experiencia sea inmersiva y técnicamente precisa.

5.3. Acciones del Usuario

En el simulador, los usuarios tienen la capacidad de interactuar con diversos elementos y realizar acciones que son esenciales para completar los experimentos. Estas interacciones están diseñadas para ser intuitivas y están respaldadas por un sistema de lógica basado en máquinas de estados y scripts específicos que controlan los flujos de acción. A continuación, se describen las principales acciones disponibles.

5.3.1. Manipulación de Objetos

El usuario puede tomar objetos interactivos, como recipientes, herramientas de laboratorio y materiales, utilizando los gestos configurados con el SDK de Oculus.

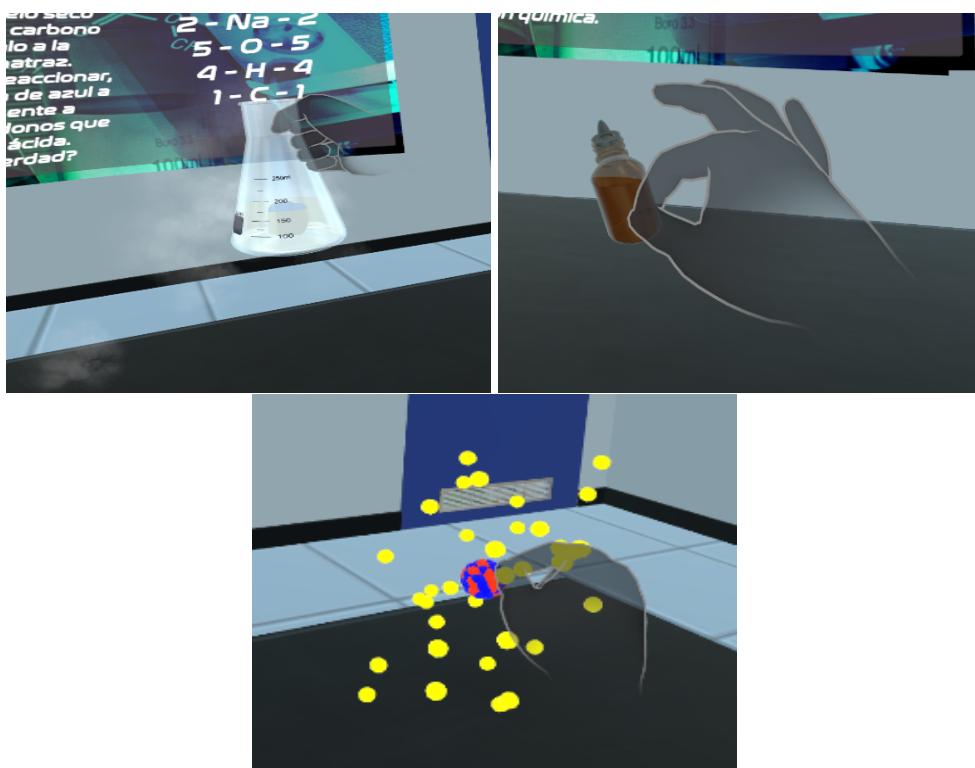


Figura 5.7: Diversas Interacciones con objetos

5.3.2. Selección de Elementos en la Tabla Periódica

El simulador incluye una tabla periódica interactiva donde el usuario puede seleccionar elementos químicos con un gesto de toque directo. Los elementos seleccionados se transforman en sus representaciones virtuales (modelos atómicos o sustancias químicas) y pueden ser llevados al área de creación o al experimento.

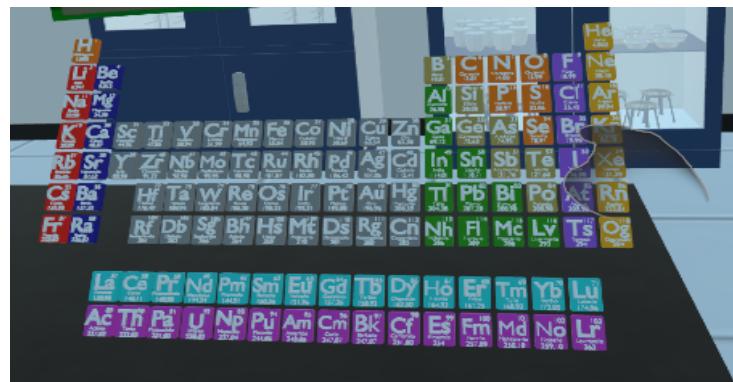


Figura 5.8: Tabla Periódica

5.3.3. Creación y Manipulación de Compuestos

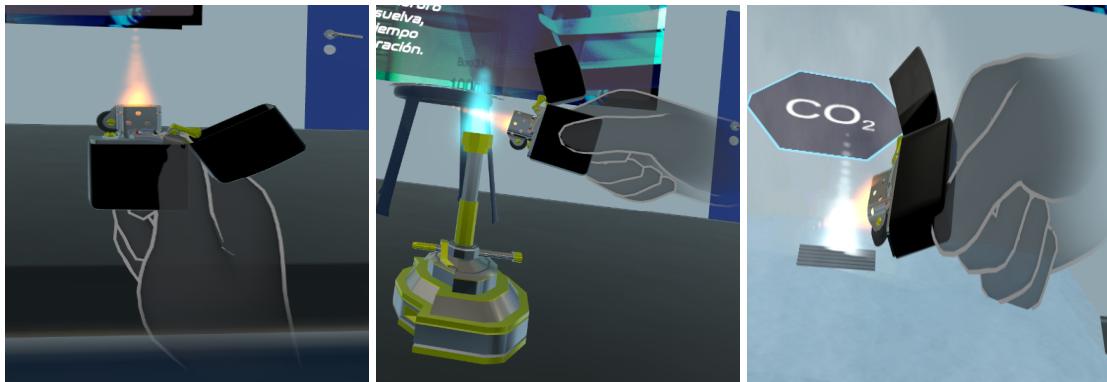
Los compuestos químicos se generan combinando elementos o materiales en áreas específicas designadas. El sistema valida automáticamente las combinaciones correctas y actualiza el estado de los experimentos, proporcionando retroalimentación visual y textual sobre el progreso.



Figura 5.9: Zona de Creación de compuestos

5.3.4. Encendido de Mecheros y Materiales

El encendido se realiza al acercar la flama del encendedor al mechero o a algún compuesto que deba quemarse. Una vez que el encendedor está lo suficientemente cerca, el sistema detecta la proximidad y activa la acción correspondiente, generando los efectos visuales y sonoros que simulan la ignición.



(a) Uso de encendedor

(b) Encendido de mechero

(c) Encendidos de material

Figura 5.10: Encendido diverso

5.3.5. Eliminación de Elementos

Para eliminar un elemento seleccionado, el usuario debe llevarlo a la zona de desecho. Una vez en esta área, el sistema lo detecta y elimina automáticamente, permitiendo mantener un espacio de trabajo ordenado y sin materiales innecesarios.

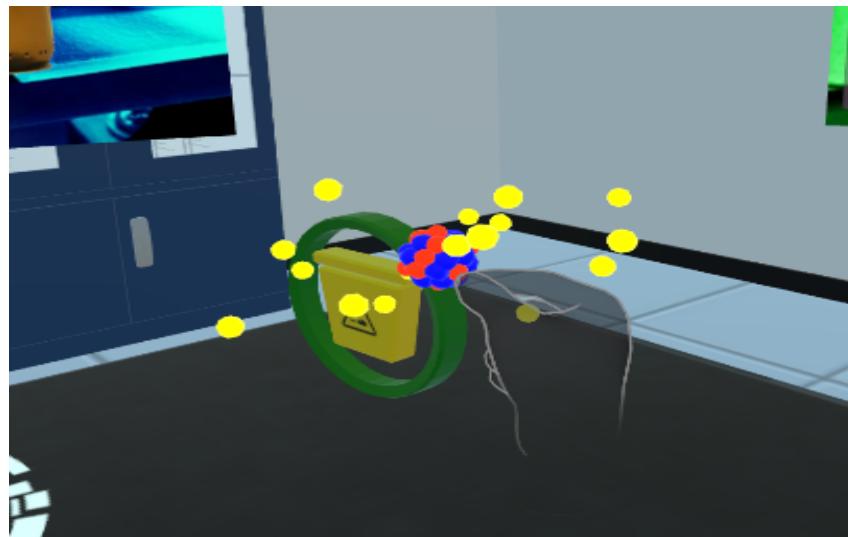


Figura 5.11: Eliminación de un elemento

5.3.6. Acciones Relacionadas con Líquidos

Las interacciones con líquidos están diseñadas para ser inmersivas y detalladas, permitiendo al usuario:

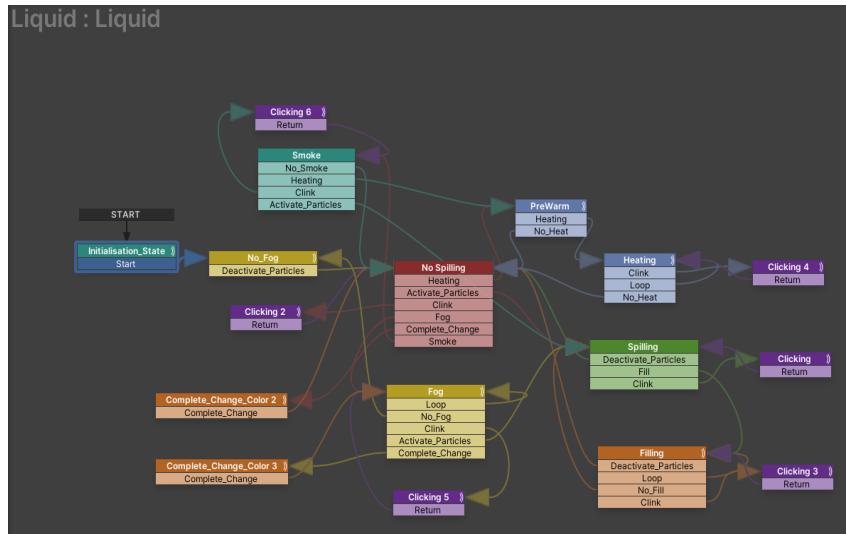
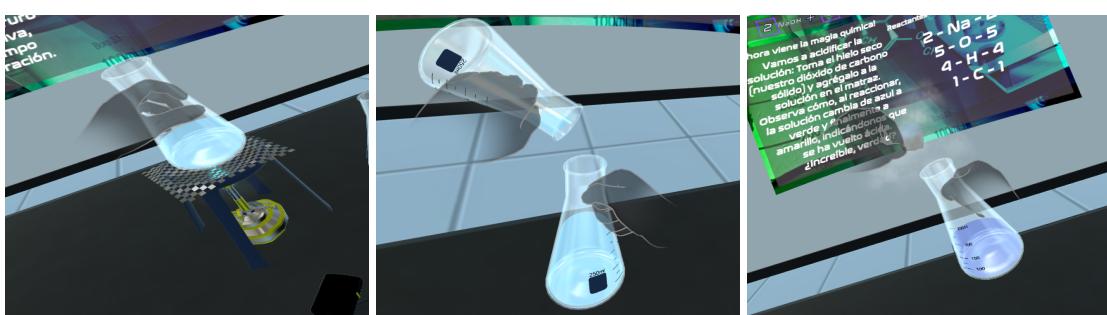


Figura 5.12: FSM Comportamiento de Liquidos

- Calentar líquidos: Al colocar un recipiente en un trípode con rejilla por encima de un mechero encendido, el líquido comienza a calentarse de manera progresiva. Este proceso está acompañado de efectos visuales como burbujeo y generación de vapor, que simulan las propiedades físicas reales del calentamiento.

El estado del calentamiento se gestiona mediante la FSM correspondiente, que controla las transiciones entre los estados como “Heating” o “PreWarm”, dependiendo de la proximidad del recipiente al mechero y el tiempo de exposición al calor (ver Figura 5.12).

- Transferencia de líquidos entre recipientes: Los usuarios pueden verter líquidos de un recipiente a otro inclinando el primero, lo que activa estados como “Spilling” o “Filling”. Estas acciones generan efectos visuales de derrame y llenado, junto con la actualización dinámica del nivel del líquido en ambos recipientes.
- Agregar sólidos a líquidos: La adición de polvos o sólidos genera reacciones químicas específicas, con cambios visuales como efervescencia o cambios de color, dependiendo del experimento.



(a) Calentamiento de Líquidos (b) Transferencia de Líquidos (c) Interacción con Sólidos

Figura 5.13: Diversas Interacciones de Líquidos

CAPÍTULO 6

Presentaciones y Pruebas del Prototipo

A lo largo del año de desarrollo, el prototipo del simulador fue presentado en diversos eventos e instituciones, lo que permitió recopilar retroalimentación clave para optimizar su funcionalidad y experiencia de usuario. Además, estas pruebas y presentaciones sirvieron para validar la precisión química de las reacciones y procesos representados, en colaboración con especialistas en el área.

6.1. Presentaciones Institucionales

El simulador fue exhibido como parte de las actividades de difusión del **Centro de Innovación y Desarrollo Tecnológico (CIDETEC) del Instituto Politécnico Nacional (IPN)**. Estas presentaciones estuvieron dirigidas a grupos invitados de diversas universidades e instituciones educativas, que tuvieron la oportunidad de observar y probar el prototipo. Entre las instituciones participantes se encuentran:

- Tecnológico de Estudios Superiores de Jocotitlán
Fecha: 24 de mayo de 2024.
- ESIME-Culhuacán, IPN
Fecha: 7 de junio de 2024.
- UPIIZ, IPN
Fechas: 21 de junio de 2024 y 27 de septiembre de 2024.
- Universidad Tecnológica de Xicotepec de Juárez
Fecha: 5 de julio de 2024.
- Universidad Mexiquense del Bicentenario
Fecha: 18 de octubre de 2024.

Estas presentaciones permitieron recopilar observaciones de estudiantes en un entorno controlado, ayudando a identificar oportunidades de mejora y ajustar detalles técnicos del simulador.



Figura 6.1: Presentaciones Institucionales

6.2. Invitaciones Externas

Además de las presentaciones institucionales, el simulador fue invitado a participar en eventos externos y actividades organizadas por otras instituciones. Estas invitaciones ampliaron el alcance del proyecto y permitieron recibir retroalimentación de públicos diversos. Los eventos destacados incluyen:

- **Centro de Estudios Superiores Isidro Fabela (Bachillerato)**
Fecha: 13 de junio de 2024.



Figura 6.2: Presentación Isidro Fabela

- **Cinvesniños, CINVESTAV - IPN**

Fechas: 22 y 23 de noviembre de 2024, en el stand “Uso de la realidad virtual en la ciencia”. Este fue el único evento público abierto, en el que participaron niños, jóvenes y adultos, ofreciendo perspectivas variadas sobre la accesibilidad y el impacto educativo del simulador.



Figura 6.3: Presentación en Cinvesniños

- **Universidad Estatal del Valle de Ecatepec**

Fecha: 27 de noviembre de 2024.



Figura 6.4: Presentación UEVE

6.3. Colaboración y Validación Académica

Durante el desarrollo del proyecto, se contó con el apoyo de la Escuela Secundaria Diurna No. 4 “Moisés Sáenz” y del profesor Ernesto Uribe. Este trabajo conjunto incluyó sesiones de prueba con estudiantes y reuniones privadas para afinar detalles técnicos y validar la precisión de las reacciones químicas y los procesos educativos mostrados en el simulador.

Las pruebas realizadas con los estudiantes ayudaron a evaluar la usabilidad y la experiencia del usuario, mientras que las reuniones con el profesor permitieron garantizar la exactitud y la relevancia de la información química representada en el sistema.



Figura 6.5: Reuniones en Secundaria 4

CAPÍTULO 7

Conclusiones

El desarrollo del prototipo de simulador de laboratorio de química inorgánica en realidad virtual permitió abordar una necesidad educativa específica mediante una herramienta tecnológica innovadora. A pesar de la existencia de otros simuladores en el ámbito de la química, muchos carecen de una experiencia inmersiva y altamente interactiva. Además, pocos están diseñados específicamente para un público mexicano, limitando su aplicabilidad y efectividad en contextos educativos locales.

El prototipo desarrollado logró implementar cuatro experimentos y un tutorial, permitiendo la realización de prácticas químicas con un nivel aceptable de precisión. Este sistema se centró en representar de manera detallada los aspectos visuales y físicos de las reacciones químicas. Dichos elementos no solo enriquecen la experiencia del usuario, sino que también contribuyen significativamente a la comprensión de los procesos químicos involucrados.

Una de las características distintivas del simulador es su capacidad para emular experimentos comúnmente realizados en aulas de nivel secundaria. Esta funcionalidad permite a los usuarios interactuar con los experimentos en un entorno virtual, ofreciendo la posibilidad de comparar sus experiencias prácticas en entornos reales. De este modo, el prototipo fomenta un aprendizaje visual, interactivo e inmersivo, que apoya el proceso de enseñanza de conceptos químicos de manera efectiva.

El desarrollo del prototipo presentó retos técnicos y multidisciplinarios, ya que involucró áreas como la química, la enseñanza, el desarrollo de software, el diseño y modelado 3D, y la implementación de tecnologías de realidad virtual. Si bien la creación de los módulos fue compleja, gran parte del esfuerzo se destinó a superar la curva de aprendizaje asociada al enfoque innovador del proyecto.

Finalmente, se concluye que los objetivos establecidos al inicio del proyecto fueron alcanzados. El prototipo cumple con el alcance planteado, al integrar un conjunto de experimentos funcionales, validados por un profesional en química, garantizando así la exactitud de los contenidos. El simulador ofrece una herramienta educativa inmersiva y funcional que puede adaptarse y escalarse en futuras iteraciones para seguir mejorando la enseñanza de la química.

CAPÍTULO 8

Trabajo a Futuro

El prototipo desarrollado presenta un punto de partida sólido para la enseñanza inmersiva de la química inorgánica. Sin embargo, se han identificado áreas clave que podrían ser abordadas en futuras iteraciones para ampliar su funcionalidad, mejorar su impacto educativo y adaptarse a un público más amplio.

I. Ampliación del Catálogo de Experimentos

Actualmente, el simulador incluye cuatro experimentos básicos. Un paso importante a futuro sería la incorporación de nuevos experimentos que abarquen conceptos avanzados de química, incluyendo experimentos que involucren química orgánica o reacciones más complejas. Esto permitiría que el simulador sea útil no solo en el nivel secundaria, sino también en niveles educativos más avanzados, como bachillerato o universidad.

II. Evaluación del Aprendizaje

Integrar un sistema de evaluación del aprendizaje proporcionaría una retroalimentación inmediata sobre el desempeño de los usuarios. Esto podría incluir cuestionarios al final de cada experimento, análisis de las decisiones tomadas durante las prácticas y la generación de reportes de progreso. Un sistema de este tipo no solo reforzaría el aprendizaje, sino que también permitiría a los docentes medir el impacto del simulador en el desarrollo de habilidades químicas en los estudiantes.

III. Compatibilidad con Diversos Equipos de RV

Para ampliar el alcance del simulador, sería importante trabajar en la compatibilidad con una mayor variedad de dispositivos de realidad virtual, como Meta Quest, HTC Vive, o dispositivos más económicos que utilizan smartphones como visores. Esta flexibilidad tecnológica permitiría que más instituciones educativas, independientemente de sus recursos, puedan implementar el simulador en sus aulas.

IV. Incorporación de Colaboración en Línea

El desarrollo de un modo multijugador o colaborativo en línea sería un avance significativo. Esto permitiría que varios usuarios trabajen juntos en experimentos en tiempo real, simulando prácticas grupales que son comunes en laboratorios físicos. Además, este enfoque fomentaría el aprendizaje cooperativo y permitiría la participación remota, haciendo el simulador accesible para estudiantes en ubicaciones geográficas distintas.

Estas propuestas de trabajo a futuro buscan consolidar el simulador como una herramienta educativa avanzada y versátil, capaz de adaptarse a diversas necesidades y contextos. Su implementación podría transformar el prototipo en una plataforma integral para el aprendizaje de la química,

APÉNDICE A

Descripción Detallada de los Experimentos

A.1. *De Básico a Ácido*

CAMBIO DE PH AL AÑADIR DIÓXIDO DE CARBONO (CO_2) A UNA SOLUCIÓN DE HIDRÓXIDO DE SODIO (NaOH) CON AZUL DE BROMOTIMOL

Introducción

El pH de una solución mide su acidez o basicidad. Un pH bajo indica que la solución es ácida, mientras que un pH alto indica que es básica. El indicador azul de bromotimol cambia de color dependiendo del pH:

- Azul en soluciones básicas ($\text{pH} > 7.6$)
- Verde en soluciones neutras ($\text{pH} \approx 7.0$)
- Amarillo en soluciones ácidas ($\text{pH} < 6.0$)

En este experimento, vamos a cambiar el pH de una solución añadiendo primero una base (NaOH) para hacerla más básica, y luego introduciremos dióxido de carbono (CO_2) para que se convierta en ácida.

Objetivo

Observar y comprender el cambio en el pH de una solución básica preparada con hidróxido de sodio (NaOH) mediante la adición de dióxido de carbono (CO_2), utilizando el indicador azul de bromotimol para identificar el cambio de básico a ácido.

Materiales y Reactivos

- Matraz Erlenmeyer (250 ml)
- 100 ml de agua destilada
- Azul de bromotimol (indicador de pH)
- Solución de hidróxido de sodio (NaOH) al 0.1 M
- Hielo seco (CO_2 sólido) o una fuente de gas CO_2
- Pipeta o gotero
- Guantes y gafas de seguridad

Procedimiento Experimental

1. Preparación de la Solución Básica:

- Vierte 100 ml de agua destilada en un Matraz Erlenmeyer.
- Agrega 2-3 gotas de azul de bromotimol. La solución debería volverse verde (pH neutro).
- Añade unas gotas de solución de NaOH (hidróxido de sodio al 0.1 M). Observa cómo el color cambia a azul, indicando que la solución es ahora básica.

Ecuación Química

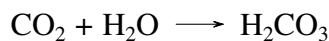


Un hidróxido se forma cuando un óxido metálico reacciona con agua.

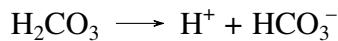
2. Acidificación de la Solución con CO₂:

Añade un pequeño trozo de hielo seco (CO₂ sólido) o introduce gas CO₂ en la solución. Observa cómo el color de la solución cambia primero a verde y luego a amarillo.

Reacción de CO₂ con Agua: El CO₂ se disuelve en agua formando ácido carbónico, lo que baja el pH:



El ácido carbónico se disocia en iones de hidrógeno (H⁺) y bicarbonto (HCO₃⁻), lo que causa la acides de la solución;



Los protones (H⁺) son responsables de la disminución del pH, lo que hace que la solución cambie a un color amarillo, indicando que ahora es ácida.

Explicación Química

En este experimento, el hidróxido de sodio (NaOH) se disuelve en agua, creando una solución básica con un pH alto, lo cual se observa como un cambio a color azul en el indicador azul de bromotimol. Luego, al añadir dióxido de carbono (CO₂), este se disuelve y reacciona con el agua para formar ácido carbónico (H₂CO₃), el cual se disocia parcialmente, liberando iones H⁺ y disminuyendo el pH de la solución.

Este cambio de pH, visible en el indicador que pasa de azul a amarillo, demuestra cómo el CO₂ puede acidificar una solución básica mediante la formación de ácido.

Resultados

Registrar el tiempo transcurrido hasta el cambio de color en cada experimento. Analizar cómo las variaciones en la concentración y la temperatura afectan este tiempo.

Conclusiones

- Se observó que, al agregar CO₂ al agua, la solución pasó de ser básica (azul) a ácida (amarillo).
- El cambio de color del indicador azul de bromotimol demostró cómo el CO₂ reduce el pH de la solución.
- Este experimento muestra cómo el dióxido de carbono puede acidificar una solución mediante la formación de ácido carbónico.

Medidas de Seguridad

- Usa guantes y gafas de seguridad para evitar el contacto con NaOH y hielo seco.
- No manipules el hielo seco directamente, usa pinzas o guantes gruesos.
- Trabaja en un área bien ventilada y no inhales el CO₂ directamente.
- Desecha los restos de la práctica de acuerdo con las normas de seguridad del laboratorio.

Cuestionario

1. ¿Qué ocurrió con el color de la solución después de añadir NaOH? ¿Por qué?
2. ¿Por qué se formó ácido carbónico al agregar CO₂ al agua?
3. ¿Qué indica el cambio de color de azul a amarillo en la solución?
4. ¿Cómo podría usar el dióxido de carbono para cambiar el pH en otros experimentos?

A.2. *La lámpara de magnesio*

COMBUSTIÓN DEL MAGNESIO EN HIELO SECO

Introducción

El magnesio es un metal altamente reactivo que puede arder incluso en ausencia de oxígeno atmosférico. En este experimento, se observará cómo el magnesio puede reaccionar con el dióxido de carbono sólido (hielo seco), un compuesto comúnmente utilizado como extintor de incendios. Esta reacción demuestra la alta posición del magnesio en la serie de reactividad, desplazando al carbono para formar óxido de magnesio y carbono elemental.

Objetivos

- Demostrar que el magnesio puede arder en ausencia de oxígeno molecular.
- Analizar los productos de la combustión del magnesio en hielo seco.
- Comprender la reactividad del magnesio frente al dióxido de carbono.

Materiales y Reactivos

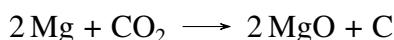
- Bloque de hielo seco (CO_2 sólido).
- Soplete o mechero Bunsen.
- Cincel o herramienta para cavidades.
- Pinzas resistentes al calor.
- Guantes térmicos.
- Gafas de seguridad.
- Pantalla de protección.
- Virutas o cinta de magnesio (aproximadamente 5 g).

Procedimiento Experimental

1. **Preparación del hielo seco:** Usa un cincel para hacer una cavidad en el bloque de hielo seco de aproximadamente 3 cm de profundidad y 3 cm de ancho. Reserva el material extraído para usarlo como tapa.
2. **Colocación del magnesio:** Introduce las virutas de magnesio en la cavidad sin llenarla completamente, dejando al menos 1 cm de espacio por encima del material.
3. **Encendido del magnesio:** Utilizando un soplete o mechero Bunsen, calienta el magnesio directamente hasta que se inicie la combustión.
Coloca rápidamente el tapón de hielo seco sobre la cavidad para minimizar la entrada de aire.
4. **Observación de la reacción:** Observa cómo el magnesio arde con una luz brillante en el ambiente de dióxido de carbono sólido, produciendo una mezcla de óxido de magnesio (polvo blanco) y carbono elemental (polvo negro).
5. **Finalización:** Permite que el residuo se enfríe antes de manipularlo.

Explicación Química

La reacción química entre el magnesio y el dióxido de carbono se puede representar con la ecuación:



El magnesio desplaza al carbono del dióxido de carbono debido a su mayor reactividad, formando óxido de magnesio (MgO) como producto principal y carbono elemental (C) como subproducto. Esto demuestra que el magnesio puede arder en condiciones donde el oxígeno no está disponible.

Resultados

Durante el experimento, se observó que el magnesio arde intensamente con una luz blanca brillante incluso en ausencia de oxígeno, utilizando el dióxido de carbono sólido como fuente de oxidación. La combustión produjo chispas y partículas expulsadas por la sublimación del hielo seco, dejando como residuo un polvo blanco (óxido de magnesio) y partículas negras (carbono elemental), evidenciando la transformación química esperada.

Conclusiones

El magnesio demostró ser un metal altamente reactivo capaz de oxidarse utilizando dióxido de carbono como agente oxidante, lo que valida su posición por encima del carbono en la serie de reactividad química. Este experimento refuerza el concepto de que el oxígeno no siempre es imprescindible para una combustión y destaca la capacidad del magnesio para desplazar al carbono, formando óxido de magnesio y carbono elemental como productos finales.

Medidas de Seguridad

- **Hielo seco:** El hielo seco debe manipularse siempre con guantes térmicos y gafas de seguridad para evitar quemaduras por congelación. Evite el contacto directo con la piel o los ojos, y no almacene el hielo seco en contenedores herméticos o espacios confinados debido al riesgo de acumulación de dióxido de carbono gaseoso que puede causar asfixia o explosión.
- **Magnesio y llama:** El magnesio y su combustión presentan riesgos significativos de quemaduras y proyección de partículas incandescentes. Utilice una pantalla de seguridad, gafas protectoras, guantes resistentes al calor y asegúrese de mantener materiales inflamables alejados del área de trabajo. Evite mirar directamente a la luz intensa producida por la combustión del magnesio para proteger sus ojos.
- **General:** Trabaje siempre en un área ventilada y mantenga al público a una distancia mínima de 2-3 metros detrás de una pantalla de seguridad. Use equipo de protección personal como bata, guantes, gafas y asegúrese de contar con extintores cerca del área de trabajo. Supervise el experimento en todo momento y siga los protocolos de manejo seguro de sustancias químicas y equipos de laboratorio.

A.3. *Nieve Química*

SÍNTESIS DE CLORURO DE AMONIO A PARTIR DE HIDRÓXIDO DE AMONIO Y ÁCIDO CLORHÍDRICO

Introducción

El cloruro de amonio (NH_4Cl) es una sal inorgánica con diversas aplicaciones industriales y científicas. En este experimento, se llevará a cabo su síntesis mediante la reacción de neutralización entre hidróxido de amonio (NH_4OH) en solución acuosa y ácido clorhídrico (HCl) en solución acuosa. La reacción es exotérmica y produce cloruro de amonio sólido, que precipita en forma de cristales.

Objetivos

- Sintetizar cloruro de amonio a partir de hidróxido de amonio y ácido clorhídrico.
- Observar la formación de cristales de cloruro de amonio como producto de la reacción.
- Comprender el concepto de reacción de neutralización y su aplicación en la síntesis de sales.

Materiales y Reactivos

- Solución de hidróxido de amonio (NH_4OH) concentrada (28-30 %)
- Solución de ácido clorhídrico (HCl) concentrada (37 %)
- Vaso de precipitados de 250 ml
- Probeta graduada de 50 ml
- Varilla de vidrio
- Pipeta graduada

Procedimiento Experimental

1. Preparación de las soluciones: medir 32.4 ml de hidróxido de amonio concentrado con una probeta graduada y verterlos en un vaso de precipitados de 250 ml.
2. Medición del ácido clorhídrico: Medir 25 ml de ácido clorhídrico concentrado con una probeta graduada.
3. Reacción de neutralización: Lentamente y con precaución, agregar el ácido clorhídrico a la solución de hidróxido de amonio. Agitar suavemente la mezcla. Observar la formación de una densa nube blanca de cloruro de amonio sólido.
4. Concentración del soluto: Calentar suavemente la solución resultante en la campana extractora hasta concentrar el soluto.
5. Enfriamiento y cristalización: Verter la solución en una probeta y enfriarla con ayuda de un ventilador. Observar la formación de cristales de cloruro de amonio en forma de estrella en la probeta.

Análisis Químico La reacción entre el hidróxido de amonio y el ácido clorhídrico es una reacción de neutralización ácido-base. El ácido clorhídrico (HCl) dona un protón (H^+) al hidróxido de amonio (NH_4OH), que actúa como base aceptando el protón. Esto resulta en la formación de agua (H_2O) y cloruro de amonio (NH_4Cl).

El cloruro de amonio formado es una sal soluble en agua, pero a medida que la solución se concentra por evaporación, se alcanza el punto de saturación y el cloruro de amonio comienza a precipitar en forma de cristales. El enfriamiento rápido de la solución favorece la formación de cristales más pequeños y numerosos, como los cristales en forma de estrella observados en el experimento.

Resultados Esperados

Reacción exotérmica: Al mezclar las soluciones de hidróxido de amonio y ácido clorhídrico, se espera observar un aumento de temperatura en el vaso de precipitados debido a la liberación de calor.

Formación de humo blanco: Al inicio de la reacción, se espera observar la formación de una densa nube blanca en el vaso de precipitados. Este humo blanco está compuesto por pequeñas partículas de cloruro de amonio sólido que se forman rápidamente en la reacción.

Precipitación de cristales: Al concentrar el soluto por evaporación y enfriar la solución, se espera observar la formación de cristales de cloruro de amonio en forma de estrella en la probeta.

Conclusiones

Este experimento permite demostrar la reacción de neutralización entre un ácido y una base, así como la formación de una sal (cloruro de amonio) como producto de la reacción. Además, ilustra la importancia de la concentración y la temperatura en la solubilidad y la cristalización de las sales. La formación de cristales de cloruro de amonio en forma de estrella es un ejemplo interesante de cómo las condiciones experimentales pueden influir en la morfología de los cristales.

A.4. *La Bruja de Bromo*

REACCIÓN DE OXIDACIÓN-REDUCCIÓN ENTRE BROMO Y ALUMINIO

Introducción

El presente experimento tiene como objetivo ilustrar la naturaleza altamente exotérmica de la reacción redox entre bromo (Br_2) y aluminio (Al), la cual resulta en la formación de bromuro de aluminio (AlBr_3). Este proceso químico es un ejemplo ilustrativo de la reactividad de los halógenos y la susceptibilidad de los metales a la oxidación.

Objetivos

- Demostrar la reacción vigorosa que ocurre entre el bromo líquido y el aluminio metálico.
- Observar la formación de bromuro de aluminio como producto de la reacción.
- Analizar los principios de oxidación y reducción en el contexto de esta reacción específica.

Materiales y Reactivos

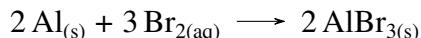
- Bromo (Br_2) líquido (¡Precaución! El bromo es altamente corrosivo y tóxico)
- Lámina de aluminio
- Tubo de ensayo
- Pipeta graduada
- Pinzas de laboratorio
- Vidrio de reloj
- Campana extractora de gases
- Guantes de nitrilo
- Gafas de seguridad

Procedimiento Experimental

1. Preparación: En el entorno controlado de una campana extractora de gases, se depositará una pequeña cantidad de bromo líquido (unas pocas gotas) en un tubo de ensayo.
2. Adición de aluminio: Utilizando pinzas de laboratorio, se introducirá con precaución un pequeño fragmento de lámina de aluminio en el tubo de ensayo que contiene el bromo.
3. Observación: Se procederá a observar detenidamente la reacción inmediata y vigorosa que se desencadena. Se espera una reacción exotérmica, acompañada de liberación de energía en forma de luz y calor. Se anticipa la formación de humo blanco correspondiente al bromuro de aluminio (AlBr_3).

Análisis Químico

La reacción química que tiene lugar entre el bromo y el aluminio se puede representar mediante la siguiente ecuación estequiométrica:



En este proceso, el aluminio experimenta oxidación, perdiendo electrones, mientras que el bromo se reduce, ganando electrones. El resultado neto es la formación de bromuro de aluminio, un sólido de color blanco.

Resultados Esperados

Se prevé observar una reacción rápida y enérgica, caracterizada por la emisión de luz y calor. La formación de humo blanco de bromuro de aluminio será un indicador visual de la ocurrencia de la reacción.

Conclusiones

Los resultados obtenidos permitirán corroborar la naturaleza altamente reactiva del bromo y la propensión del aluminio a la oxidación. La reacción servirá como modelo para ilustrar los principios fundamentales de las reacciones redox y la transferencia electrónica entre especies químicas.

Medidas de Seguridad

- ¡Precaución Extrema! El bromo es una sustancia altamente corrosiva y tóxica. Su manipulación debe realizarse exclusivamente en una campana extractora de gases y utilizando equipo de protección personal adecuado, incluyendo guantes de nitrilo y gafas de seguridad.
- Es imperativo evitar el contacto directo con el bromo. En caso de contacto accidental con la piel, se debe lavar inmediatamente la zona afectada con abundante agua y buscar atención médica si es necesario.
- La inhalación de los vapores de bromo debe ser evitada en todo momento.
- El experimento debe ser realizado bajo la supervisión de personal debidamente capacitado y experimentado en el manejo de sustancias químicas peligrosas.
- Es fundamental contar con un extintor de incendios adecuado y estar familiarizado con los protocolos de emergencia en caso de cualquier eventualidad.

APÉNDICE B

Scripts Relevantes

Este apéndice contiene los scripts más relevantes utilizados en la implementación del simulador. Cada uno de ellos desempeña un papel específico en la lógica, interacción o control de las escenas del simulador. A continuación, se presenta una breve descripción de cada script.

B.1. MenuController.cs

Este script se encarga de gestionar la lógica principal del menú del **Hub**, permitiendo al usuario seleccionar entre el tutorial y los experimentos. También interactúa con la máquina de estados finita (FSM) para manejar las transiciones entre escenas y actualiza la interfaz de usuario con los datos correspondientes.

```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using UnityEngine;
5  using UnityEngine.UI;
6  using TMPro;
7
8  public class MenuController : MonoBehaviour
9  {
10     private Text Tittle;
11     private TextMeshProUGUI Data;
12     private Image Reference;
13     [SerializeField] private PlayMakerFSM fsm;
14     private string experiment = "Experiment_";
15     private ExperimentData dataSelected;
16
17     public void assign_fields(GameObject tittle, GameObject data,
18                               GameObject reference)
19     {
20         if(tittle == null)
21         {
22             Debug.LogError("Tittle");
23         }
24         Tittle = tittle.GetComponent<Text>();
25         if(data == null)
26         {
```

```
26             Debug.LogError("Data");
27         }
28         Data = data.GetComponent<TextMeshProUGUI>();
29         if(reference == null)
30         {
31             Debug.LogError("Reference");
32         }
33         Reference = reference.GetComponent<Image>();
34     }
35
36     public void SelectedData(ExperimentData selected)
37     {
38         if (selected != null)
39         {
40             dataSelected = selected;
41         }
42         else
43         {
44             Debug.LogError("Selected data is null.");
45         }
46     }
47
48     public void change_Data()
49     {
50         if(Tittle != null && Data != null && Reference != null &&
51             → dataSelected != null)
52         {
53             Tittle.text = dataSelected.getName;
54             Data.text = dataSelected.getDescription;
55             Reference.sprite = dataSelected.getReference;
56         }
57         else
58         {
59             Debug.LogError("Missing references or dataSelected is
60             → null.");
61         }
62     }
63
64     public enum FSMEvent
65     {
66         Tutorial_Data,
67         Experiment_1_Data,
68         Experiment_2_Data,
69         Experiment_3_Data,
70         Experiment_4_Data,
71         Return_Hub,
72         Return_Experiments_Data,
73         Experiments_Data
74     }
75
76     private FSMEvent GetFSMEventByIndex(int index)
77     {
78         FSMEvent[] values =
79             → (FSMEvent[])Enum.GetValues(typeof(FSMEvent));
80     }
81 }
```

```
77
78     if (index == 6)
79     {
80         int aux = dataSelected.getName == "Tutorial" ? 5 : 6;
81         return values[aux];
82     }
83
84     if (index >= 0 && index < values.Length)
85     {
86         return values[index]; // Devuelve el valor
87         ↵ correspondiente al índice
88     }
89     else
90     {
91         Debug.LogError("Índice fuera de rango");
92         return FSMEvent.Return_Hub;
93     }
94
95 // Función para enviar un evento al FSM directamente usando el
96 // enum
96 public void SendFSMEventByIndex(int index)
97 {
98     FSMEvent eventToSend = GetFSMEventByIndex(index);
99     if (fsm != null)
100    {
101        Debug.Log(eventToSend.ToString());
102        fsm.SendEvent(eventToSend.ToString());
103    }
104 }
105
106 public void change_scene()
107 {
108     string name_scene = dataSelected.getName == "Tutorial" ?
109         ↵ "Tutorial" : experiment+dataSelected.getNo_Experiment;
110
111     fsm.SendEvent(name_scene);
112 }
```

B.2. Experiment_1_Manager.cs

Este script controla la lógica específica del primer experimento, incluyendo la secuencia de sustancias esperadas y las transiciones en la FSM según el progreso del usuario. También gestiona cambios visuales, como el color de los objetos, en función del estado del experimento.

```

1  using System.Collections.Generic;
2  using UnityEngine;
3
4  public class Experiment_1_Manager : MonoBehaviour, ISubstanceManager
5  {
6      [SerializeField] private List<string> substanceSequence; // 
    ↳ Secuencia esperada de sustancias
7      [SerializeField] private List<string> fsmEvents; // Eventos FSM
    ↳ asociados a la secuencia
8      [SerializeField] private string fsmName; // Nombre del FSM
    ↳ específico
9      [SerializeField] private List<Color> targetColors; // Colores
    ↳ para cada estado
10     [SerializeField] private List<float> lerpDurations; // Duraciones
    ↳ de interpolación para cada estado
11     private PlayMakerFSM fsm;
12     private Dictionary<SubstanceTracker, int> trackerStates = new
    ↳ Dictionary<SubstanceTracker, int>();
13
14     private void Awake()
15     {
16         // Obtener el FSM en el mismo objeto por nombre
17         var fsmArray = GetComponents<PlayMakerFSM>();
18         fsm = System.Array.Find(fsmArray, f => f.FsmName == fsmName);
19         if (fsm == null)
20         {
21             Debug.LogWarning($"FSM con el nombre '{fsmName}' no
    ↳ encontrado en el mismo objeto.");
22         }
23     }
24     public void RegisterTracker(SubstanceTracker tracker)
25     {
26         if (!trackerStates.ContainsKey(tracker))
27         {
28             trackerStates[tracker] = 0;
29         }
30     }
31
32     public void OnSubstanceAdded(SubstanceTracker tracker, string
    ↳ substance)
33     {
34         if (!trackerStates.ContainsKey(tracker)) return;
35
36         int currentState = trackerStates[tracker];
37         if (currentState < substanceSequence.Count && substance ==
    ↳ substanceSequence[currentState])
38         {

```

```
39         trackerStates[tracker]++;
40         TriggerFSMEvent(tracker, fsmEvents[currentState]);
41
42         if (currentState > 0 && currentState <=
43             targetColors.Count)
44         {
45             ChangeColorForState(tracker, currentState - 1); // 
46             ↵ Ajusta el índice para coincidir con la lista
47         }
48     }
49
50     private void TriggerFSMEvent(SubstanceTracker tracker, string
51         ↵ eventName)
52     {
53         if (fsm != null)
54         {
55             fsm.SendEvent(eventName);
56             Debug.Log($"Event '{eventName}' triggered for tracker
57             ↵ '{tracker.name}'");
58         }
59     }
60
61     private void ChangeColorForState(SubstanceTracker tracker, int
62         ↵ stateIndex)
63     {
64         GameObject trackerObject = tracker.gameObject;
65         ColorChangeController colorController =
66             ↵ trackerObject.GetComponent<ColorChangeController>();
67
68         if (colorController != null)
69         {
70             ↵ colorController.StartColorChange(targetColors[stateIndex],
71             ↵ lerpDurations[stateIndex]);
72             Debug.Log($"Cambio de color iniciado para
73             ↵ '{trackerObject.name}' con el color
74             ↵ {targetColors[stateIndex]} y duración
75             ↵ {lerpDurations[stateIndex]}.");
76         }
77         else
78         {
79             Debug.LogWarning($"El objeto '{trackerObject.name}' no
80             ↵ tiene un componente ColorChangeController.");
81         }
82     }
83 }
```

B.3. Experiment_2_Manager.cs

El gestor del segundo experimento valida las combinaciones de sustancias introducidas por el usuario y maneja las interacciones relacionadas con la temperatura, como el calentamiento y enfriamiento de los materiales. Utiliza la FSM para desencadenar eventos según el comportamiento de los reactivos.

```

1  using System.Collections.Generic;
2  using UnityEngine;
3  using System.Linq;
4
5  public class Experiment_2_Manager : MonoBehaviour, ISubstanceManager
6  {
7      [SerializeField] private List<string> validCombination; // Lista
     → de sustancias esperadas para una mezcla
8      [SerializeField] private string fsmName; // Nombre del FSM
     → específico
9      private PlayMakerFSM fsm;
10     private Dictionary<SubstanceTracker, HashSet<string>>
     → trackerSubstances = new Dictionary<SubstanceTracker,
     → HashSet<string>>();
11     private Dictionary<SubstanceTracker, float> boilingTimeCounters =
     → new Dictionary<SubstanceTracker, float>();
12     private Dictionary<SubstanceTracker, bool> boilingFlags = new
     → Dictionary<SubstanceTracker, bool>();
13     private float boilingDelay = 10f;
14
15     private void Awake()
16     {
17         // Obtener el FSM en el mismo objeto por nombre
18         var fsmArray = GetComponents<PlayMakerFSM>();
19         fsm = System.Array.Find(fsmArray, f => f.FsmName == fsmName);
20         if (fsm == null)
21         {
22             Debug.LogWarning($"FSM con el nombre '{fsmName}' no
     → encontrado en el mismo objeto.");
23         }
24     }
25     public void RegisterTracker(SubstanceTracker tracker)
26     {
27         if (!trackerSubstances.ContainsKey(tracker))
28         {
29             trackerSubstances[tracker] = new HashSet<string>();
30         }
31         // Inicializar el contador de tiempo para este tracker
32         if (!boilingTimeCounters.ContainsKey(tracker))
33         {
34             boilingTimeCounters[tracker] = 0f;
35         }
36
37         if (!boilingFlags.ContainsKey(tracker))
38         {
39             boilingFlags[tracker] = false;

```

```
40         }
41     }
42
43     public void OnSubstanceAdded(SubstanceTracker tracker, string
44         ↳ substance)
45     {
46
47         if (!trackerSubstances.ContainsKey(tracker)) return;
48
49         // Agregar la sustancia al conjunto de sustancias del tracker
50         trackerSubstances[tracker].Add(substance);
51
52         // Comprobar si la combinación es válida
53         if (IsCombinationValid(tracker.GetSubstances()))
54         {
55
56             fsm.SendEvent("goCalentamiento");
57             PlayMakerFSM playMakerFSM =
58                 ↳ tracker.gameObject.GetComponent<PlayMakerFSM>();
59             playMakerFSM.SendEvent("Smoke");
60         }
61     }
62
63     private bool IsCombinationValid(List<string> substances)
64     {
65
66         // Validar estrictamente
67         bool isValid = substances.Count == validCombination.Count
68             && !substances.Except(validCombination).Any()
69             && !validCombination.Except(substances).Any();
70
71         return isValid;
72     }
73
74
75     private void HandleTemperature(float temperature, float
76         ↳ boilingPoint, SubstanceTracker tracker)
77     {
78
79         // Realizar acciones dependiendo de la temperatura
80         if (temperature > boilingPoint && !boilingFlags[tracker])
81         {
82
83             // Incrementar el contador de tiempo específico para este
84             ↳ tracker
85             boilingTimeCounters[tracker] += Time.deltaTime;
86
87             if (boilingTimeCounters[tracker] >= boilingDelay)
88             {
89                 fsm.SendEvent("goEnfriamiento");
90                 boilingFlags[tracker] = true;
91             }
92         }
93         else if (temperature < boilingPoint)
94         {
95
96             PlayMakerFSM playMakerFSM =
97                 ↳ tracker.gameObject.GetComponent<PlayMakerFSM>();
98             boilingTimeCounters[tracker] -= Time.deltaTime;
99
100        }
```

```
88         if (boilingTimeCounters[tracker] <= 0 &&
89             → boilingFlags[tracker])
90         {
91             fsm.SendEvent ("goExplicacion");
92         }
93     }
94
95 private void FixedUpdate()
96 {
97     // Comprobar la temperatura de cada tracker de manera
98     // periódica
99     foreach (var tracker in trackerSubstances.Keys)
100    {
101        TemperatureController temperatureController =
102            → tracker.gameObject.GetComponent<TemperatureController>();
103        if (temperatureController != null &&
104            → (temperatureController.GetFlag() ||
105            → boilingFlags[tracker]))
106        {
107            // Comprobar la temperatura actual y ejecutar las
108            // acciones basadas en la temperatura
109            → HandleTemperature(temperatureController.GetCurrentTemperature(),
110                → temperatureController.GetBoilingPoint(),
111                → tracker);
112        }
113    }
114}
```

B.4. Experiment_3_Manager.cs

Este script se enfoca en validar combinaciones químicas en el tercer experimento, gestionando además el nivel de líquido en el recipiente. Supervisa cambios en el shader asociado al material del objeto y dispara eventos en la FSM basándose en el progreso del usuario.

```

1  using System.Collections.Generic;
2  using UnityEngine;
3  using System.Linq;
4
5  public class Experiment_3_Manager : MonoBehaviour, ISubstanceManager
6  {
7      [SerializeField] private List<string> validCombination; // Lista
    ↵ de sustancias esperadas para una mezcla
8      [SerializeField] private string fsmName; // Nombre del FSM
    ↵ específico
9      [SerializeField] private Renderer objectRenderer;
10     private float level;
11     private bool level_reached = false;
12     private PlayMakerFSM fsm;
13     private Dictionary<SubstanceTracker, HashSet<string>>
    ↵ trackerSubstances = new Dictionary<SubstanceTracker,
    ↵ HashSet<string>>();
14
15     private void Awake()
16     {
17         // Obtener el FSM en el mismo objeto por nombre
18         var fsmArray = GetComponents<PlayMakerFSM>();
19         fsm = System.Array.Find(fsmArray, f => f.FsmName == fsmName);
20         if (fsm == null)
21         {
22             Debug.LogWarning($"FSM con el nombre '{fsmName}' no
    ↵ encontrado en el mismo objeto.");
23         }
24     }
25
26     void FixedUpdate()
27     {
28         if (!level_reached)
29             levelTarget();
30     }
31
32     public void RegisterTracker(SubstanceTracker tracker)
33     {
34         if (!trackerSubstances.ContainsKey(tracker))
35         {
36             trackerSubstances[tracker] = new HashSet<string>();
37         }
38     }
39
40     public void OnSubstanceAdded(SubstanceTracker tracker, string
    ↵ substance)
41     {

```

```
42     if (!trackerSubstances.ContainsKey(tracker)) return;
43
44     // Agregar la sustancia al conjunto de sustancias del tracker
45     trackerSubstances[tracker].Add(substance);
46
47     // Comprobar si la combinación es válida
48     if (IsCombinationValid(tracker.GetSubstances()))
49     {
50         fsm.SendEvent("Reaccion_Exotermica");
51         PlayMakerFSM playMakerFSM =
52             → tracker.gameObject.GetComponent<PlayMakerFSM>();
53         playMakerFSM.SendEvent("Smoke");
54     }
55 }
56
57 private bool IsCombinationValid(List<string> substances)
58 {
59     // Validar estrictamente
60     bool isValid = substances.Count == validCombination.Count
61         && !substances.Except(validCombination).Any()
62         && !validCombination.Except(substances).Any();
63
64     return isValid;
65 }
66
67 private void levelTarget()
68 {
69     level = objectRenderer.material.GetFloat("_Fill");
70     if(level >= -0.045f)
71     {
72         fsm.SendEvent("next");
73         level_reached = true;
74     }
75 }
```

B.5. Experiment_4_Manager.cs

El cuarto gestor valida combinaciones químicas específicas para el experimento final.

Supervisa la interacción del usuario con las sustancias y utiliza la FSM para avanzar en el flujo del experimento.

```

1  using System.Collections.Generic;
2  using UnityEngine;
3  using System.Linq;
4
5  public class Experiment_4_Manager : MonoBehaviour, ISubstanceManager
6  {
7      [SerializeField] private List<string> validCombination; // Lista
    ↳ de sustancias esperadas para una mezcla
8      [SerializeField] private string fsmName; // Nombre del FSM
    ↳ específico
9      private PlayMakerFSM fsm;
10     private Dictionary<SubstanceTracker, HashSet<string>>
    ↳ trackerSubstances = new Dictionary<SubstanceTracker,
    ↳ HashSet<string>>();
11
12     private void Awake()
13     {
14         // Obtener el FSM en el mismo objeto por nombre
15         var fsmArray = GetComponents<PlayMakerFSM>();
16         fsm = System.Array.Find(fsmArray, f => f.FsmName == fsmName);
17         if (fsm == null)
18         {
19             Debug.LogWarning($"FSM con el nombre '{fsmName}' no
    ↳ encontrado en el mismo objeto.");
20         }
21     }
22
23     public void RegisterTracker(SubstanceTracker tracker)
24     {
25         if (!trackerSubstances.ContainsKey(tracker))
26         {
27             trackerSubstances[tracker] = new HashSet<string>();
28         }
29     }
30
31     public void OnSubstanceAdded(SubstanceTracker tracker, string
    ↳ substance)
32     {
33         if (!trackerSubstances.ContainsKey(tracker)) return;
34
35         // Agregar la sustancia al conjunto de sustancias del tracker
36         trackerSubstances[tracker].Add(substance);
37
38         // Comprobar si la combinación es válida
39         if (IsCombinationValid(tracker.GetSubstances()))
40         {
41             fsm.SendEvent("next");

```

```

42         }
43     }
44
45     private bool IsCombinationValid(List<string> substances)
46     {
47         // Validar estrictamente
48         bool isValid = substances.Count == validCombination.Count
49             && !substances.Except(validCombination).Any()
50             && !validCombination.Except(substances).Any();
51
52         return isValid;
53     }
54 }
```

B.6. SubstanceTracker.cs

Este script gestiona el seguimiento de sustancias en el simulador, permitiendo detectar interacciones entre objetos y registrar las sustancias que contiene cada objeto. También puede compartir sustancias entre objetos y conectarse con el gestor correspondiente a la escena activa.

```

1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4   using UnityEngine.SceneManagement;
5
6   public class SubstanceTracker : MonoBehaviour
7   {
8       // Lista de tags que representan las sustancias contenidas en
9       // este objeto
10      [SerializeField] private List<string> containedSubstances = new
11          List<string>();
12      [SerializeField] private LayerMask allowedLayers;
13      [SerializeField] private PlayMakerFSM fsm;
14      [SerializeField] private Vector3 startPosition;
15      [SerializeField] private Vector3 targetPositionOffset; // Offset
16          // para la posición del hermano
17      [SerializeField] private Vector3 targetRotationEuler;
18      private ISubstanceManager manager;
19
20      private void Awake()
21      {
22          FindAndConnectManager();
23          // Verifica si el objeto tiene un tag asignado antes de
24          // llamar a AddSubstance
25          if (gameObject.tag != "Untagged")
26          {
27              AddSubstance(gameObject.tag);
28          }
29      }
30
31      private void FindAndConnectManager()
```

```
28     {
29         // Determinar el tipo de manager según la escena actual
30         string currentSceneName = SceneManager.GetActiveScene().name;
31
32         switch (currentSceneName)
33         {
34             case "Experiment_1":
35                 manager = FindObjectOfType<Experiment_1_Manager>();
36                 break;
37             case "Experiment_2":
38                 manager = FindObjectOfType<Experiment_2_Manager>();
39                 break;
40             case "Experiment_3":
41                 manager = FindObjectOfType<Experiment_3_Manager>();
42                 break;
43             case "Experiment_4":
44                 manager = FindObjectOfType<Experiment_4_Manager>();
45                 break;
46             default:
47                 Debug.LogError($"No se encontró un manager en la
48                             ↵ escena '{currentSceneName}'.");
49                 break;
50
51         if (manager != null)
52         {
53             ConnectToManager(manager); // Conexión al manager
54         }
55     }
56
57     // Método para agregar una nueva sustancia a la lista
58     public void AddSubstance(string substanceTag)
59     {
60         if (!containedSubstances.Contains(substanceTag))
61         {
62             containedSubstances.Add(substanceTag);
63             // Notificar al manager cuando se añade una nueva
64             ↵ sustancia
65             manager?.OnSubstanceAdded(this, substanceTag);
66         }
67     }
68
69     // Método para compartir sustancias con otro objeto
70     public void ShareSubstances(SubstanceTracker targetTracker)
71     {
72         foreach (string substance in containedSubstances)
73         {
74             targetTracker.AddSubstance(substance);
75         }
76     }
77
78     // Devuelve la lista de sustancias contenidas
79     public List<string> GetSubstances()
```

```

79     {
80         return containedSubstances;
81     }
82
83     // Detecta interacciones con objetos en el Layer permitido usando
84     // un Trigger
85     private void OnTriggerEnter(Collider other)
86     {
87         // Verificar si el objeto pertenece al Layer permitido
88         if ((allowedLayers.value & (1 << other.gameObject.layer)) !=
89             0)
90         {
91             AddSubstance(other.gameObject.tag);
92
93             // Cambiar el objeto colisionado a hermano
94             MakeObjectSibling(other.gameObject);
95
96             if (fsm != null)
97             {
98                 fsm.SendEvent("Fog");
99             }
100        }
101
102        // Método para conectar este tracker con un manager externo
103        public void ConnectToManager<T>(T manager) where T :
104            ISubstanceManager
105        {
106            manager.RegisterTracker(this);
107
108        // Método para hacer que el objeto colisionado sea hermano del
109        // objeto actual
110        private void MakeObjectSibling(GameObject otherObject)
111        {
112            Transform originalParent = otherObject.transform.parent;
113
114            originalParent.SetParent(transform.parent);
115
116            // Desactivar el Rigidbody y los Colliders de originalParent
117            Rigidbody rb = originalParent.GetComponent<Rigidbody>();
118            if (rb != null)
119            {
120                rb.isKinematic = true; // Desactivar la física, si es
121                // necesario
122                rb.useGravity = false; // Desactivar la gravedad
123            }
124
125            // Desactivar todos los colliders en el originalParent
126            Collider[] colliders =
127                originalParent.GetComponentsInChildren<Collider>();
128            foreach (Collider col in colliders)
129            {

```

```
126         if (!col.isTrigger) // Verifica si el collider no es un
127             ↵ trigger
128         {
129             col.enabled = false;
130         }
131
132         // Definir la rotación antes de la interpolación de posición
133         Quaternion targetRotation =
134             ↵ Quaternion.Euler(targetRotationEuler); // Convertir los
135             ↵ valores serializados a Quaternion
136         originalParent.localRotation = targetRotation;
137         Debug.Log(originalParent.localRotation);
138
139         // Realizar la transición de posición usando una corutina
140             ↵ para un movimiento suave
141         StartCoroutine(ChangePositionSmoothly(originalParent, 3f));
142             ↵ // 3 segundo de transición
143     }
144
145     // Corutina para hacer un cambio suave de posición
146     private IEnumerator ChangePositionSmoothly(Transform
147         ↵ originalParent, float transitionDuration)
148     {
149
150         //Vector3 startPosition = originalParent.localPosition;
151
152         float timeElapsed = 0f;
153
154         // Realizar la interpolación hasta que el tiempo de
155             ↵ transición se haya completado
156         while (timeElapsed < transitionDuration)
157         {
158
159             // Interpolamos la posición de manera suave
160             ↵ originalParent.localPosition =
161                 ↵ Vector3.Lerp(startPosition, targetPositionOffset,
162                     ↵ timeElapsed / transitionDuration);
163
164             timeElapsed += Time.deltaTime;
165             yield return null; // Esperamos hasta el siguiente frame
166         }
167
168         // Aseguramos que la posición final sea exactamente la
169             ↵ targetPositionOffset
170         originalParent.localPosition = targetPositionOffset;
171     }
172 }
```

B.7. Liquid_Level.cs

Este script controla el nivel de líquido dentro de un recipiente. Ajusta dinámicamente el nivel en función de los ángulos de inclinación del objeto, simulando derrames o llenado. También activa y desactiva partículas asociadas con el comportamiento del líquido, y se integra con la FSM para manejar eventos.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using HutongGames.PlayMaker; // Importa la API de PlayMaker
5
6  public class Liquid_Level : MonoBehaviour
7  {
8      // Referencia al Renderer del GameObject para manipular
9      // → propiedades del shader
10     [SerializeField] private Renderer objectRenderer;
11     [SerializeField] private float start_level;
12     [SerializeField] private float max_angle;
13     [SerializeField] private float min_level;
14     [SerializeField] private float max_level;
15     [SerializeField] private float m_level;
16     [SerializeField] private GameObject container;
17     [SerializeField] private GameObject particles; // Referencia al
18     // → sistema de partículas
19     [SerializeField] private PlayMakerFSM fsm;
20     private float level; // Nivel actual del agua
21     private float new_level; // Nuevo nivel que se ajusta en el
22     // → shader
23     private float spillAngle; // Ángulo en el que comienza el derrame
24     private float spillSpeed = 0.01f; // Velocidad base de vaciado
25     // → por grado de diferencia
26     private float spillRate; // Tasa de derrame
27     private float xAngle, zAngle; // Ángulos de inclinación en los
28     // → ejes X y Z
29     private Fill_detection fill_detection; // Referencia al script de
30     // → detección de llenado
31
32     // Se llama cuando el objeto es activado
33     private void OnEnable()
34     {
35         // Obtener el Renderer para controlar el nivel del agua en el
36         // → shader
37         objectRenderer = GetComponent<Renderer>();
38         objectRenderer.material.SetFloat("_Fill", start_level); // → Inicializar el nivel de agua
39         new_level = objectRenderer.material.GetFloat("_Fill");
40         fill_detection = particles.GetComponent<Fill_detection>(); // → Obtener el script de detección de llenado
41     }
42
43     // Se ejecuta cada frame fijo
44     void FixedUpdate()
45     {

```

```

38     level = objectRenderer.material.GetFloat("_Fill"); // 
    ↵ Actualizar el nivel del agua
39     spilling(); // Llamar al método de derrame
40 }
41
42 // Controla el derrame del agua basado en el ángulo de
    ↵ inclinación
43 void spilling ()
44 {
45     // Obtener los ángulos X y Z, ajustándolos al rango de -180 a
    ↵ 180
46     xAngle = Mathf.DeltaAngle(0f,
    ↵ container.transform.eulerAngles.x);
47     zAngle = Mathf.DeltaAngle(0f,
    ↵ container.transform.eulerAngles.z);
48 // Si el nivel del agua está por encima del mínimo permitido
49 if (level > min_level)
50 {
51     spillAngle = CalculateSpillAngle(level); // Calcular el
    ↵ ángulo en que ocurre el derrame
52     // Verificar si el ángulo actual en X o Z supera el
    ↵ ángulo de derrame
53     if (IsSpilling(spillAngle))
54     {
55         // Calcular la diferencia entre el ángulo actual y el
    ↵ ángulo de derrame
56         float angleDifference =
            ↵ Mathf.Abs(Mathf.DeltaAngle(CalculateCompositeAngle(),
            ↵ spillAngle));
57         spillRate = spillSpeed * angleDifference; // Calcular
            ↵ la tasa de derrame
58         // Reducir el nivel de agua según la tasa de derrame
59         downLevel(spillRate);
60         fsm.SendEvent("Activate_Particles");
61         fill_detection.detection(spillRate); // Detectar
            ↵ llenado de otros objetos
62     }
63     else
64     {
65         // Si no hay derrame, desactivar las partículas
66         fsm.SendEvent("Deactivate_Particles");
67     }
68 }
69 else
70 {
71     // Asegurar que el nivel no baje por debajo del mínimo y
    ↵ desactivar partículas
72     objectRenderer.material.SetFloat("_Fill", min_level);
73     fsm.SendEvent("Deactivate_Particles");
74 }
75 }
76
77 public void downLevel(float downRate)

```

```

78     {
79         if (level > min_level)
80         {
81             // Reducir el nivel de agua según la tasa de vaciado
82             new_level -= downRate * Time.fixedDeltaTime / 25f;
83             objectRenderer.material.SetFloat("_Fill", new_level); // → Asignar el nuevo nivel al shader
84         }
85     else
86     {
87         // Asegurar que el nivel no baje por debajo del mínimo y
88         // → desactivar partículas
89         objectRenderer.material.SetFloat("_Fill", min_level);
90     }
91
92     // Método que aumenta el nivel de agua según la tasa de llenado
93     public void filling_out(float fillRate)
94     {
95         if (level < max_level + 0.01f) // Limitar el nivel de llenado
96         {
97             new_level += fillRate * Time.fixedDeltaTime / 25f; // → Incrementar el nivel del agua
98             objectRenderer.material.SetFloat("_Fill", new_level); // → Actualizar el nivel en el shader
99         }
100    }
101
102    // Calcula el ángulo a partir del cual comienza el derrame,
103    // → basado en el nivel del agua
104    private float CalculateSpillAngle(float level)
105    {
106        if (level == min_level)
107        {
108            return 90f; // Si está vacío, el matraz no se derrama
109        }
110        else if (level < m_level)
111        {
112            return max_angle; // Con niveles bajos, el ángulo es
113            // → mayor
114        }
115        else if (level >= max_level)
116        {
117            return 1f; // Si está lleno, el derrame ocurre de
118            // → inmediato
119        }
120    else
121    {
122        // Interpolación lineal entre ángulos de derrame según el
123        // → nivel
124        float normalizedFactor = (level - m_level) / (max_level +
125        // → 0.1f);
126        return Mathf.Lerp(max_angle, 0f, normalizedFactor);

```

```
122         }
123     }
124
125     // Verifica si el ángulo actual está por encima del ángulo de
126     // derrame
127     private bool IsSpilling(float spillAngle)
128     {
129         // Calcula la diferencia compuesta de los ángulos X y Z con
130         // el ángulo de derrame
131         float compositeAngle = CalculateCompositeAngle();
132         return compositeAngle > spillAngle || compositeAngle > 360f -
133             spillAngle;
134     }
135
136     private float CalculateCompositeAngle()
137     {
```

Bibliografía

- [1] A. Brito, «Nuevas coordenadas para la alfabetización: debates, tensiones y desafíos en el escenario de la cultura digital,» UNESCO IIEP Buenos Aires. Oficina para América Latina y Organización de Estados Iberoamericanos para la Educación, la Ciencia y la Cultura, Buenos Aires, 2015. dirección: <https://unesdoc.unesco.org/ark:/48223/pf0000371034.locale=es> (visitado 14-09-2023).
- [2] J. Trejo-Quintana, «La falta de acceso y aprovechamiento de los medios y las tecnologías: dos deudas de la educación en México,» en *Educación y pandemia. Una visión académica*, México, 2020, págs. 122-128. dirección: <https://www.iisue.unam.mx/nosotros/covid/educacion-y-pandemia> (visitado 14-09-2023).
- [3] I. Pastor Sánchez, *Análisis de la Metodología STEM a través de la percepción docente*, 2018. dirección: <http://uvadoc.uva.es/handle/10324/30952> (visitado 15-10-2023).
- [4] T. Zamorano Escalona, Y. García Cartagena y D. Reyes González, «Educación para el sujeto del siglo XXI: principales características del enfoque STEAM desde la mirada educacional,» *Contextos: Estudios de Humanidades Y Ciencias Sociales*, n.º 41, 14 de sep. de 2018. dirección: <http://revistas.umce.cl/index.php/contextos/article/view/1386> (visitado 17-09-2023).
- [5] D. Zapatero Guillén, «La realidad virtual como recurso y herramienta útil para la docencia y la investigación,» *TEyET*, n.º 6, págs. 17-23, 1 de dic. de 2011.
- [6] J. L. Hernández, A. Violant, M. I. Binimelis y J. L. Gutiérrez, «Química,» en *alfa nauta - Programa Educativo Temático*, Ediciones Nauta C., S.A., abr. de 1997, págs. 171-289, ISBN: 84-89140-51-0.
- [7] G. C. Lizeth, «Stoichiometry,» *PREPA3*, vol. 10, n.º 19, págs. 58-59, ene. de 2023.
- [8] H. E. Avery, *Basic Reaction Kinetics and Mechanisms*. The Macmillan Press Ltd, 1974, ISBN: 978-0-333-15381-9. DOI: 10.1007/978-1-349-15520-0.
- [9] F. García Mata, *Leyes Ponderales*, PDF, 2021. dirección: https://drive.google.com/file/d/1knPRAuxNInB-RMNJuajqGzBiA4q_qQcb/view.
- [10] R. Coss Bú, *Simulación - Un enfoque práctico*. Editorial Limusa S.A. de C.V., 2003, ISBN: 968-18-1506-8.
- [11] J. Jerald, *The VR Book: Human-Centered Design for Virtual Reality*. ACM Books, 2016, ISBN: 978-1-97000-112-9.

- [12] T. Ortiz, K. Nathalia, D. C. H. Muñoz y W. N. M. Mendoza., «Importancia de los laboratorios remotos y virtuales en la educación superior,» *wpecbti*, vol. 1, n.º 1, ago. de 2020. doi: 10.22490/ECBTI.3976..
- [13] Autodesk Support. «Documentación de Maya.» (2023), dirección: <https://www.autodesk.es/support/technical/article/caas-tsarticles/tsarticles/ESP/ts/1C3jaffqnWFyQoLPEPm7n.html> (visitado 29-04-2024).
- [14] Blender Foundation. «Blender Documentation.» (2024), dirección: <https://docs.blender.org/> (visitado 29-04-2024).
- [15] Maxon. «Maxon ZBrush Documentation.» (2024), dirección: <https://help.maxon.net/zbr/en-us/> (visitado 29-04-2024).
- [16] Adobe Inc. «Adobe Illustrator User Guide.» (2024), dirección: <https://helpx.adobe.com/mx/illustrator/user-guide.html> (visitado 03-05-2024).
- [17] GIMP Team. «GIMP Documentation.» (2024), dirección: <https://www.gimp.org/docs/> (visitado 03-05-2024).
- [18] Inkscape Community. «Inkscape Documentation.» (2024), dirección: <https://gitlab.com/inkscape/inkscape-docs/documentation> (visitado 03-05-2024).
- [19] Quantitative Software Management. «Function Point Languages Table.» (2024), dirección: <https://www.qsm.com/resources/function-point-languages-table> (visitado 25-05-2024).