



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

“UN JUEGO SERIO PARA MEJORAR EL PROCESO DE ENSEÑANZA/APRENDIZAJE DE LA IDENTIFICACIÓN DE *STAKEHOLDERS* EN LA ELICITACIÓN DE REQUISITOS DE *SOFTWARE*: UN CASO DE ESTUDIO ENFOCADO A ESTUDIANTES DE PREGRADO”

TESIS

**PARA OBTENER EL TÍTULO DE
INGENIERA EN COMPUTACIÓN**

PRESENTA

C. LUISA ANDREA MORALES GARCÍA

DIRECTOR DE TESIS

DRA. CARLA LENINCA PACHECO

CODIRECTOR DE TESIS

DR. IVÁN ANTONIO GARCÍA PACHECO

HUAJUAPAN DE LEÓN, OAX. A 02 DE OCTUBRE DE 2023

Tesis presentada el 02 de octubre de 2023

ante los siguientes sinodales:

M.C. Gerardo Cruz González

Ing. David Martínez Torres

M.C. Ricardo Ruíz Rodríguez

Directora de Tesis:

Dra. Carla Lenín Pacheco Agüero

Co-director:

Dr. Iván Antonio García Pacheco

Dedicatoria

Con cariño a mi madre, padre y familia, gracias por todo su apoyo y amor, no solo durante estos años de mi carrera, sino por todo el tiempo que me han apoyado y han visto por mí.

Agradecimientos

Por medio de estas líneas quiero agradecer:

A mi directora y co-director de Tesis, la Dra. Carla Leninca Pacheco Agüero y el Dr. Iván Antonio García Pacheco, por su guía y paciencia, y por darse el tiempo de dirigirme y ayudarme a realizar un trabajo excelente.

A los profesores que conforman el comité revisor y evaluador de este trabajo de Tesis, por el tiempo que dedicaron a su revisión.

A mis compañeros egresados de la carrera en Ingeniería de Computación de la Universidad Tecnológica de la Mixteca, por su participación en la evaluación empírica del juego creado.

A mi profesor Azael Trujillo, gracias por toda su ayuda, apoyo, revisiones, realimentación y, sobre todo, por su tiempo.

A mi mamá, Rafaela García Sánchez, y a mi padre, Luis Octavio Morales Blánhir, por su apoyo incondicional, por su cariño, por siempre ver por mí y quererme.

A Daniel Enrique Guzmán Jiménez, por ser un buen amigo y soporte durante la carrera, siempre estaré muy agradecida contigo, gracias por todo.

Índice

Índice	ix
Lista de tablas	xv
Lista de figuras	xvii
Resumen	xix
1. Introducción.....	1
1.1. Contexto del problema.....	1
1.2. Importancia del problema.....	4
1.3. Necesidad de resolución	6
1.4. Delimitaciones de la Tesis.....	9
1.5. Hipótesis del trabajo	9
1.6. Objetivos del trabajo.....	9
1.6.1. Objetivo general	9
1.6.2. Objetivos específicos.....	9
1.7. Aproximación a la solución.....	10
1.8. Evaluación empírica	12
1.9. Estructura preliminar de la Tesis.....	13
2. Marco teórico.....	15
2.1. La importancia de la Ingeniería de Requisitos en la industria de <i>software</i>	15
2.2. Problemas dentro de la industria	17
2.3. Enseñanza de la Ingeniería de Requisitos en estudiantes de pregrado	20
2.4. Estado del arte	22
2.4.1. Juegos serios	22
2.4.2. La aplicación de los juegos serios dentro de la Ingeniería de <i>Software</i>	26
2.4.3. El uso de la gamificación en la enseñanza/aprendizaje de la Ingeniería de Requisitos	29
2.4.3.1. DMGame+: Explorando el conocimiento sobre la Ingeniería de Requisitos con gamificación	29
2.4.3.1.1. Objetivo	29
2.4.3.1.2. Descripción.....	29
2.4.3.1.3. Resultados obtenidos	31

2.4.3.2. Agon: Un marco de trabajo basado en la gamificación para la Ingeniería de Requisitos	31
2.4.3.2.1. Objetivo	31
2.4.3.2.2. Descripción	32
2.4.3.2.3. Resultados	36
2.4.4. El uso de los juegos serios en la enseñanza/aprendizaje de la elicitación de requisitos	37
2.4.4.1. El aprendizaje activo de la elicitación de requisitos de <i>software</i> con LEGO	37
2.4.4.1.1. Objetivo	37
2.4.4.1.2. Descripción	37
2.4.4.1.3. Resultados	39
2.4.4.2. Diseño y desarrollo de un juego serio para la enseñanza de la elicitación y el análisis de requisitos	40
2.4.4.2.1. Objetivo	40
2.4.4.2.2. Descripción	41
2.4.4.2.3. Resultados	43
2.4.4.3. Desarrollo de un videojuego educativo para la enseñanza/aprendizaje de la clasificación de requisitos en la Ingeniería de <i>Software</i>	44
2.4.4.3.1. Objetivo	44
2.4.4.3.2. Descripción	44
2.4.4.3.3. Resultados	46
2.4.5. El uso de la gamificación y juegos serios en la enseñanza/aprendizaje de la identificación de los <i>stakeholders</i> dentro de la Ingeniería de Requisitos	47
2.4.5.1. GARUSO: Una plataforma de gamificación para involucrar a los <i>stakeholders</i> en un contexto organizacional	47
2.4.5.1.1. Objetivo	47
2.4.5.1.2. Descripción	47
2.4.5.1.3. Resultados	50
2.4.5.2. <i>Requengin</i> : Un juego serio para enseñar los fundamentos del estándar ISO/IEC/IEEE 29148	52
2.4.5.2.1. Objetivo	52
2.4.5.2.2. Descripción	52
2.4.5.2.3. Resultados	54
2.5. Consideraciones finales sobre el estado del arte	55
3. Metodología	57
3.1. Pre-1: Diseño de los retos educativos, las competencias básicas y los objetivos educativos	59
3.2. Pre-2: Diseño del tipo de juego	62
3.2.1. Género	63
3.2.2. Plataforma	63
3.2.3. Propósito y público objetivo	63
3.2.4. Área de la aplicación	63
3.3. Pre-3: Diseño inicial de la historia y personajes principales	63

3.3.1. Flujo del juego	64
3.3.2. Personajes	70
3.3.2.1. Detective	70
3.3.2.2. Policía encargado del caso	70
3.3.2.3. Alcalde de <i>Wonderland</i>	70
3.3.2.4. Experta forense	70
3.3.2.5. Testigo	70
3.3.2.6. Secretaria particular del alcalde	70
3.3.2.7. Periódicos del día del accidente y cinco días anteriores	70
3.3.2.8. Cinta de video del día del atentado	71
3.3.2.9. Fisgón	71
3.3.2.10. Informante anónimo	71
3.3.2.11. NPC	71
3.4. Fase 1: Diseño del capítulo	71
3.5. Fase 2: Diseño de la escena	72
3.5.1. C1_1_Tutorial	73
3.5.1.1. Diseño del escenario	74
3.5.1.2. Personajes	75
3.5.2. C1_2_Escoger_Avatar	75
3.5.2.1. Diseño del escenario	75
3.5.2.2. Personajes	75
3.5.3. C2_0_Entrada_Ciudad	75
3.5.3.1. Diseño del escenario	75
3.5.3.2. Personajes	76
3.5.4. C2_1_Casa_Víctima	76
3.5.4.1. Diseño del escenario	76
3.5.4.2. Personajes	76
3.5.5. C2_2_Morgue	77
3.5.5.1. Diseño del escenario	77
3.5.5.2. Personajes	77
3.5.6. C2_3_Estación_Trenes	77
3.5.6.1. Diseño del escenario	77
3.5.6.2. Personajes	78
3.5.7. C2_4_Cuarto_Cámaras	78
3.5.7.1. Diseño del escenario	78
3.5.7.2. Personajes	78
3.5.8. C2_5_Parque	78
3.5.8.1. Diseño del escenario	78
3.5.8.2. Personajes	79

3.5.9. C2_6_Parque de juegos	79
3.5.9.1. Diseño del escenario	79
3.5.9.2. Personajes	79
3.5.10. C2_7_Límites_Ciudad.....	79
3.5.10.1. Diseño del escenario	79
3.5.10.2. Personajes	79
3.5.11. C2_8_Área_Residencial	79
3.5.11.1. Diseño del escenario	79
3.5.11.2. Personajes	80
3.5.12. C2_9_Final_Perdiste.....	80
3.5.12.1. Diseño del escenario	80
3.5.12.2. Personajes	80
3.5.13. C2_10_Final_Moriste.....	80
3.5.13.1. Diseño del escenario	80
3.5.13.2. Personajes	80
3.5.14. C2_11_Final_Ganaste.....	80
3.5.14.1. Diseño del escenario	80
3.5.14.2. Personajes	80
3.5.15. C2_12_Plantilla_De_Volere.....	80
3.5.15.1. Diseño del escenario	81
3.6. Fase 3: Identificación/etiquetado de los retos educativos y evaluación	81
3.6.1. Identificar y consultar todas las posibles fuentes de requisitos (<i>stakeholders</i>)	81
3.6.2. Identificar clases de usuarios y sus características	81
3.6.3. Identificar y consultar con los actores del sistema	82
3.6.4. Utilizar esquemas para caracterizar y evaluar las relaciones apropiadas entre todos los <i>stakeholders</i>	82
3.7. Fase 4: Identificación/clasificación de emociones	82
3.7.1. Emociones presentes en la escena C1_1_Tutorial.....	82
3.7.2. Emociones presentes en la escena C1_2_Escoger_Avatar	82
3.7.3. Emociones presentes en la escena C2_0_Entrada_Ciudad.....	83
3.7.4. Emociones presentes en la escena C2_1_Casa_Víctima	83
3.7.5. Emociones presentes en la escena C2_2_Morgue	83
3.7.6. Emociones presentes en la escena C2_3_Estación_Trenes	83
3.7.7. Emociones presentes en la escena C2_4_Cuarto_Cámaras.....	84
3.7.8. Emociones presentes en la escena C2_5_Parque.....	84
3.7.9. Emociones presentes en la escena C2_6_Parque de juegos	84
3.7.10. Emociones presentes en la escena C2_7_Límites_Ciudad.....	84
3.7.11. Emociones presentes en la escena C2_8_Área_Residencial	84
3.7.12. Emociones presentes en la escena C2_9_Final_Perdiste.....	84
3.7.13. Emociones presentes en la escena C2_10_Final_Moriste.....	85

3.7.14. Emociones presentes en la escena C2_11_Final_Ganaste	85
3.7.15. Emociones presentes en la escena C2_12_Plantilla_De_Volere.....	85
3.8. Fase 5: Diseño de la adaptación	85
3.9. Fase 6: Diseño de la colaboración	85
3.10. Implementación del videojuego.....	86
3.10.1. Herramientas y lenguaje de programación	86
3.10.2. Desarrollo	86
3.10.3. Elementos de apoyo extras dentro del videojuego	94
4. Evaluación empírica	97
4.1. Estrategia de evaluación	97
4.2. Participantes	97
4.3. Instrumentos	98
4.4. Desarrollo del caso de estudio	101
4.4.1. Resultados obtenidos por el grupo de control	102
4.4.2. Resultados obtenidos por el grupo experimental.....	103
4.4.3. Evaluación de la efectividad del juego	106
4.4.3.1. Evaluación de la usabilidad del juego	107
4.4.3.2. Evaluación de la confianza de lo(a)s estudiantes	108
4.4.3.3. Evaluación del desafío del juego	109
4.4.3.4. Evaluación de la satisfacción de lo(a)s estudiantes	110
4.4.3.5. Evaluación de la diversión dentro del juego.....	110
4.4.3.6. Evaluación de la atención enfocada de lo(a)s estudiantes	111
4.4.3.7. Evaluación de la relevancia del juego	112
4.4.3.8. Evaluación del aprendizaje percibido por lo(a)s estudiantes.....	112
4.4.3.9. Consideraciones finales sobre los resultados obtenidos en la evaluación empírica	114
4.4.3.10. Recomendaciones y sugerencias del grupo experimental para mejorar el juego	114
5. Conclusiones	117
6. Bibliografía.....	119
7. Anexo A.- Acrónimos	119

Lista de tablas

Tabla 1. Mapeo de personajes del juego con las clases de <i>Volere</i>	62
Tabla 2. Retos educativos y forma de evaluación dentro del juego serio	81
Tabla 3. Cuestionario diseñado para evaluar los conocimientos de los estudiantes.....	98
Tabla 4. Instrumento de medición de MEEGA+ (adaptado de Petri et al., (2021)).....	99
Tabla 5. Percepciones de los estudiantes sobre la usabilidad del juego	107
Tabla 6. Percepciones de los estudiantes sobre la confianza en el juego	109
Tabla 7. Percepciones de los estudiantes sobre el desafío del juego.....	109
Tabla 8. Percepciones de los estudiantes sobre la satisfacción con el juego.....	110
Tabla 9. Percepciones de los estudiantes sobre la diversión del juego.....	111
Tabla 10. Percepciones de los estudiantes sobre su atención enfocada en el juego	111
Tabla 11. Percepciones de los estudiantes sobre la relevancia del juego	112
Tabla 12. Percepciones de los estudiantes sobre el aprendizaje experimentado con el juego	113
Tabla 13. Percepciones de los estudiantes sobre la cobertura de los objetivos de aprendizaje.....	113
Tabla 14. Recomendaciones y sugerencias de los estudiantes del grupo experimental respecto al diseño del juego	114

Lista de figuras

Figura 1.1. Metodología de desarrollo de juegos basada en guiones interactivos (adaptada de Larios et al., (2015) y de Lope et al., (2015))	11
Figura 2.1. Problemas clave en la Ingeniería de Requisitos y su criticidad (tomada de Méndez (2018))	20
Figura 2.2. Interfaz de usuario de <i>DMGame+</i> (tomada de Perini et al., (2018))	30
Figura 2.3. Obtención de un metamodelo de requisitos con <i>Agon-Tool</i> (tomada de Piras (2018))	32
Figura 2.4. Elicitación y análisis de requisitos con <i>Agon-Tool</i> (tomada de Piras (2018))	33
Figura 2.5. Registro de requisitos con <i>Agon-Tool</i> (tomada de Piras (2018))	33
Figura 2.6. Caracterización del contexto con <i>Agon-Tool</i> (tomada de Piras (2018))	34
Figura 2.7. Análisis del contexto de los requisitos con <i>Agon-Tool</i> (tomada de Piras (2018))	34
Figura 2.8. Refinamiento de los requisitos con <i>Agon-Tool</i> (tomada de Piras (2018))	35
Figura 2.9. Uso de la gamificación para operacionalizar los requisitos con <i>Agon-Tool</i> (tomada de Piras (2018))	35
Figura 2.10. Instanciación de la gamificación con <i>Agon-Tool</i> (tomada de Piras (2018))	36
Figura 2.11. Instrucciones recibidas para la realización de la misión (tomada de Ibrahim et al., (2019))	42
Figura 2.12. Información del <i>stakeholder</i> y el trabajo que desempeña (tomada de Ibrahim et al., (2019))	43
Figura 2.13. Interfaz propuesta para el juego creado con <i>GDevelop</i> (tomada de Chanchí et al., (2019))	45
Figura 2.14. Interfaz definitiva del juego para clasificar requisitos funcionales y no funcionales (tomada de Chanchí et al., (2019))	46
Figura 2.15. Pantalla inicial de GARUSO (tomada de Kolpondinos y Glinz (2020))	48
Figura 2.16. Formato utilizado en GARUSO para compartir un <i>post</i> (tomada de Kolpondinos y Glinz (2020))	49
Figura 2.17. Esquema de visualización de recompensas en GARUSO (tomada de Kolpondinos y Glinz (2020))	51
Figura 2.18. Pantalla de inicio de <i>Requengin</i> (tomada de García et al., (2020))	52
Figura 2.19. Uso de la técnica de entrevistas en <i>Requengin</i> (tomada de García et al., (2020))	53
Figura 2.20. Generación de casos de uso en <i>Requengin</i> (tomada de García et al., (2020))	54

Figura 3.1. Metodología de desarrollo de juegos basada en guiones interactivos y las diferentes grandes áreas de conocimiento comprendidas en el desarrollo de un juego (adaptada de Larios et al., (2015) y de Lope et al., (2015)).....	57
Figura 3.2. Clases de <i>stakeholders</i> (traducida de Robertson y Robertson (2012)).....	62
Figura 3.3. Pantalla de inicio del juego “ <i>Misterio en Wonderland</i> ”	64
Figura 3.4. Ejemplo del tutorial para explicar la movilidad durante el juego	65
Figura 3.5. Ejemplo del tutorial para explicar la interacción con personajes y objetos	65
Figura 3.6. Pantalla para escoger el avatar	66
Figura 3.7. Ejemplo de interacción del(a) jugador(a) con el entorno	66
Figura 3.8. Ejemplo de interacción para encontrar el video de la cámara.....	67
Figura 3.9. Ejemplo de inicio de plática entre el(la) jugador(a) y algún personaje.....	68
Figura 3.10. Flujo de las escenas entre capítulos del juego.....	74
Figura 3.11. Pantalla de inicio del juego serio.....	86
Figura 3.12. Pantalla de carga animada	87
Figura 3.13. Ejemplo del tutorial del juego	87
Figura 3.14. Avatares disponibles en el juego.....	88
Figura 3.15. Inicio del juego en la ciudad de <i>Wonderland</i>	89
Figura 3.16. Ejemplo de diálogos en entrevistas y/o interrogatorios	89
Figura 3.17. Ampliación del mapa de la ciudad de <i>Wonderland</i>	90
Figura 3.18. Búsqueda de información en la casa de la víctima.....	90
Figura 3.19. Identificación de documentos en el juego	91
Figura 3.20. Inventario de objetos recogidos en el juego	91
Figura 3.21. Búsqueda de pistas en el cuarto de cámaras.....	92
Figura 3.22. Estación de trenes.....	92
Figura 3.23. Lista de sospechosos	93
Figura 3.24. Resumen de diálogos.....	94
Figura 3.25. Resumen de misión actual.....	94
Figura 3.26. Elementos disponibles en la tienda	95
Figura 3.27. Menú de pausa.....	95
Figura 4.1. Estudiantes tomando la clase teórica como parte de la evaluación.....	101
Figura 4.2. Evaluación del desempeño académico correspondiente al grupo de control	102
Figura 4.3. Estudiantes del grupo experimental realizando el tutorial del juego	104
Figura 4.4. Estudiantes del grupo experimental durante la sesión de juego.....	105
Figura 4.5. Evaluación del desempeño académico correspondiente al grupo experimental	106

Resumen

Numerosos proyectos de *software* inconclusos o cancelados pueden rastrearse, en gran parte, a una mala identificación y relación con los *stakeholders* del proyecto. De acuerdo con datos de la industria, esto puede atribuirse a la poca o casi nula enseñanza, a nivel de pregrado (i.e., licenciatura), de la elicitación de requisitos de *software*, a pesar de que es un proceso crucial de la Ingeniería de Requisitos, ya que es donde los requisitos preliminares deben quedar definidos. Es por lo que, en este trabajo de Tesis, se propone el desarrollo y uso de “*Misterio en Wonderland*”, un juego serio que permite mejorar el aprendizaje en estudiantes de licenciatura sobre la identificación de los *stakeholders* involucrados en un proyecto de *software*. Dicho juego tiene como actividad única la identificación de *stakeholders* a través de la plantilla de *Volere*. Por último, se realizó una evaluación empírica sobre la efectividad del juego para cumplir esta tarea que incluyó a estudiantes de noveno semestre de la carrera de Ingeniería de Computación de la Universidad Tecnológica de la Mixteca. Finalmente, se proporcionan los resultados obtenidos a través de dicha evaluación.

1. Introducción

1.1. Contexto del problema

Piattini (2016), Ebert (2018), y Sommerville (2019) argumentan que la Ingeniería de *Software* (IS) surgió durante la conferencia impartida por la Organización del Tratado del Atlántico Norte (OTAN) en 1968 en Alemania, como una opción para resolver la crisis del *software*. Para Brooks (1987) y Gibbs (1994) esta crisis se refería a los inconvenientes en el desarrollo de *software* que provocaban que la entrega se retrasara, se sobrepasara el presupuesto, no se cumpliera con los requisitos establecidos, y que, por ende, fuera más difícil de controlar.

De acuerdo con Garousi et al. (2019), la IS puede ayudar a resolver los problemas previamente mencionados disminuyendo la complejidad del *software*, reduciendo costos, minimizando los tiempos de entrega con el objetivo de desarrollar *software* confiable; sin embargo, esto no sucede la mayoría de las veces, ya que de acuerdo con los datos del *Chaos Report*¹, mostrados en Johnson (2020), únicamente el 30% de los proyectos de *software* fueron terminados con éxito, lo que significa, que fueron entregados a tiempo con las características, funciones y requisitos requeridos, y dentro del presupuesto.

Desde el surgimiento de la IS han existido diferentes definiciones para dicho término, entre las cuales destacan las siguientes:

- Sommerville (2019) afirma que se trata de la ingeniería involucrada en todos los procesos que conlleva la realización de un *software* y se encuentra presente desde la definición hasta el mantenimiento de éste, una vez que se pone en funcionamiento.
- De acuerdo con el Cuerpo de Conocimiento de la Ingeniería de *Software* (SWEBOK, por sus siglas en inglés), la IS es una disciplina emergente cuyo nivel de madurez se encuentra en aumento y puede considerarse como una aplicación para el desarrollo, procedimiento y mantenimiento de un *software* (Bourque y Fairley, 2014).
- El estándar ISO/IEC/IEEE:29148 considera que es “la aplicación sistemática de conocimientos, métodos y experiencia científicos y tecnológicos al diseño, implementación, prueba y documentación del *software*” (ISO/IEC/IEEE, 2018).

¹ El *Chaos Report* es un informe llevado a cabo anualmente por *The Standish Group* sobre los proyectos de tecnologías de la información, con el objetivo de proporcionar información sobre el grado de éxito que alcanzó un proyecto al término de su desarrollo, a la par de proporcionar informes sobre cuáles fueron las posibles causas de estos resultados (Johnson, 2018).

En lo que respecta a su estructura, Sommerville (2019) y Pressman y Maxim (2019) coinciden en que la IS está conformada por cinco procesos:

1. Análisis y especificación de requisitos.
2. Diseño.
3. Implementación.
4. Validación y verificación.
5. Mantenimiento.

Hussain et al., (2016) afirmaron que la primera etapa es esencial para que el proyecto tenga éxito, ya que si se pasa por alto la inclusión de una correcta Ingeniería de Requisitos (IR) se tendrá más probabilidad de tener fallos y de correr otros riesgos consecuentes. Ya que lo que se busca hacer dentro de esta fase es recopilar, comprender, y especificar los requisitos² proporcionados por los *stakeholders*³ para que éstos guíen el desarrollo de un *software* de calidad. La IR es considerada de suma importancia ya que cualquier error que se cometa aquí puede provocar inconvenientes en las etapas posteriores.

Al ser esta etapa de gran importancia es necesario saber su significado y en qué consiste. Entre las definiciones más utilizadas para la IR se encuentran las siguientes:

- Dick et al. (2017) argumentaron que es “el subconjunto de la Ingeniería de *Software* que se ocupa de descubrir, desarrollar, rastrear, analizar, calificar, comunicar y gestionar los requisitos que definen el sistema en niveles sucesivos de abstracción”.
- El estándar ISO/IEC/IEEE:29148 considera que se trata de una disciplina que se encarga de proporcionar y mantener los requisitos que debe de realizar un *software* para lo cual es necesario investigar, obtener, desarrollar, examinar, y rectificar los requisitos (ISO/IEC/IEEE, 2018).

Al igual que la IS, la IR está conformada por diferentes procesos. De acuerdo con Hussain et al., (2016) y Aberkane et al., (2021) los principales son los siguientes:

1. Elicitación.
2. Análisis.
3. Especificación.
4. Validación y verificación.
5. Gestión.

En este sentido, Wiegers y Beatty (2013) argumentan que la Elicitación de Requisitos (ER) se encarga de obtener las necesidades y limitaciones de los *stakeholders*, paso previo para la definición de los requisitos. La ER se lleva a cabo con la intención de definir no sólo requisitos de los

² Para el estándar ISO/IEC/IEEE:29148 (2018) un requisito es “una declaración que traduce o expresa una necesidad y sus restricciones asociadas y condiciones”.

³ Los *stakeholders* son aquellos grupos o individuos (e.g., empleados, clientes, proveedores, accionistas, bancos, ambientalistas, gobierno, etc.) que están interesados en el *software* o que tienen conocimiento relacionado con éste y que pueden afectarlo directamente (Freeman y McVea, 2005; Robertson y Robertson, 2012).

stakeholders, sino también de recopilar requisitos empresariales, funcionales y no funcionales. En la elicitación, los *stakeholders* son un elemento crítico, ya que proporcionan los requisitos, además al involucrarlos en el proyecto se facilita la validación del mismo. La importancia de esta etapa radica en la propia naturaleza de la IR, ya que el desarrollo de los requisitos es un problema cíclico, en donde como primer paso se lleva a cabo la elicitación, como segundo paso se analiza la información recabada y, por último, se escriben los requisitos donde, en caso de que haga falta información, se podrá realizar una elicitación adicional y repetir los pasos.

La ER es una etapa crítica en el desarrollo del *software* ya que cualquier error en ésta podría provocar desvíos significativos en la producción. Estos problemas se deben principalmente a la intervención del factor humano (i.e., *stakeholders*), ya que no tienden a categorizar ni estructurar sus deseos y necesidades de forma correcta, lo que convierte a la ER en una fase compleja (Becker et al., 2019).

De acuerdo con Loucopoulos y Karakostas (1995), Bourque y Fairley (2014), y el Modelo de Capacidad y Madurez Integrado para el Desarrollo 2.0 (CMMI-DEV 2.0, por sus siglas en inglés) (CMMI, 2018), las actividades que comprende la ER son cuatro:

1. Identificación de los *stakeholders*.
2. Aplicación de la técnica de elicitación de requisitos.
3. Obtención de los requisitos de los *stakeholders*.
4. Refinamiento de la información elicitada.

El SWEBOK (Bourque y Fairley, 2014) afirma de hecho que los *stakeholders* deben ser correcta y completamente identificados en la etapa de la elicitación de requisitos, para que lo(a)s ingeniero(a)s de requisitos puedan trabajar en conjunto con ellos y así poder elicitar los requisitos. Es así como una incorrecta identificación y asignación de importancia a los *stakeholders* puede asociarse al fracaso en los proyectos de *software*. Por lo que es necesario que el ingeniero(a) de requisitos identifique, ordene y administre la importancia y los puntos de vista de todos los *stakeholders*, de igual forma, deberá mediar entre el conocimiento del dominio por parte de los *stakeholders* y la IR. En cuanto a la identificación de los *stakeholders*, este cuerpo de conocimiento menciona que el tipo de *stakeholder* a identificar depende del tipo de proyecto asignado, además, es necesario prestar atención a esta actividad ya que no necesariamente van a ser grupos homogéneos, ya que cada uno de ellos va a tener diferentes deseos y necesidades, por lo que en ocasiones se necesitará la búsqueda de *stakeholders* que sean externos a la organización, como reguladores o autoridades.

Carrizo y Rojas (2018) analizaron dos investigaciones basadas en artículos, así como encuestas realizadas a ingeniero(a)s de *software* que trabajaban en la industria, para concluir que una de las razones por las que un proyecto no era exitoso en la industria de *software*, es una inapropiada elicitación y especificación de los requisitos. Esto derivado de los cambios frecuentes en las necesidades y deseos de los *stakeholders*, las restricciones contractuales, así como las actualizaciones tecnológicas. Considerando lo anterior, los autores propusieron desarrollar un marco de trabajo que permitiera obtener la información del proyecto a desarrollar y poder así identificar mejor a los *stakeholders* y que éstos definan, por ende, mejor sus deseos y necesidades.

Por otro lado, Becker et al., (2019) señalaron que el éxito de un proyecto de *software* debía ser un asunto de preocupación para las empresas de *software*, ya que para que un proyecto sea exitoso es necesario que se cumplan los requisitos establecidos por los *stakeholders*, a la par de que se cumplan con las limitaciones como el tiempo y los costos propuestos, para lo cual hay que haber identificado completa y correctamente a los *stakeholders*.

Como se puede observar con las investigaciones anteriormente expuestas, la identificación de los *stakeholders* es fundamental no nada más en la ER, sino durante todo el proceso de desarrollo de *software*. Sin embargo, es también uno de los grandes problemas dentro de la industria ya que rara vez se identifican correcta y completamente los *stakeholders*, por lo que no se establece adecuadamente lo que se espera del producto final y esto ocasiona que se produzcan proyectos fracasados (Carrizo y Rojas, 2018).

En el *Chaos Report* del 2018 (Johnson, 2018), por ejemplo, quedaron de manifiesto dos factores que hacen que un proyecto sea considerado como exitoso y que tienen que ver con la identificación y participación de los *stakeholders*:

1. Falta de participación e involucramiento de los *stakeholders*.
2. Requisitos incompletos.

Lamentablemente, la identificación de los *stakeholders* dentro de la ER no ha recibido la importancia y la atención que se merece ya que en las diferentes iniciativas de la IS como el SWEBOK (Bourque y Fairley, 2014), CMMI-DEV v2.0 (2018) e ISO/IEC/IEEE: 29148 (2018), solo se proveen ejemplos numerosos de quiénes pueden ser los *stakeholders*, dejando de lado las prácticas y estrategias que ayuden a lo(a)s ingeniero(a)s de requisitos a identificarlos correcta y completamente.

No obstante, hay una excepción entre estas iniciativas: la plantilla de *Volere* (Robertson y Robertson, 2012). Esta plantilla es un marco detallado que permite identificar a los *stakeholders* que están participando en el desarrollo de un sistema a través de su clasificación en base a clases, características y tipo de conocimiento (Pacheco y Tovar, 2007; Martins et al., 2013).

En base a lo anterior, se puede afirmar que, para la elicitación de requisitos, la identificación de los *stakeholders* es una actividad clave.

1.2. Importancia del problema

Como se expuso en la sección anterior, es necesario que dentro de la IR se implementen estrategias que garanticen el éxito en el desarrollo del *software*, específicamente en lo que se refiere a la elicitación de requisitos. Sin embargo, uno de los mayores problemas que se presentan es la brecha de interpretación que se da entre el ingeniero(a) de requisitos y los *stakeholders*, dando como resultado errores de identificación e interpretación de sus deseos y necesidades (Becker et al., 2019).

Por ejemplo, en la investigación de De la Cruz y Castro (2015) se citó una frase de Maté y Silva (2005) que afirma que “la calidad del *software* depende de la calidad de los requisitos elicitados”. Así mismo, Wong (2019) afirma que, en los últimos años, los problemas presentados durante la elicitación de requisitos se han vuelto más frecuentes, provocando así una deficiencia en la obtención de “buenos”⁴ requisitos.

Oliveros y Martínez (2015) argumentaron que los *stakeholders* representan puntos clave no solo en la elicitación de requisitos, sino en el desarrollo y éxito de un proyecto de *software*, ya que son éstos quienes proporcionan las funcionalidades y restricciones que deberá cumplir el *software*, además, de que son necesarios para su validación. Por lo que sugieren que involucrar a los *stakeholders* en la elicitación de requisitos significa entender sus necesidades. Sin embargo, para poder realizar este entendimiento es necesario una comprensión y un involucramiento adecuado del

⁴ De acuerdo con Coventry (2015), un buen requisito es aquel que especifica una funcionalidad que es necesaria, verificable y alcanzable, así como también cumple con que es claro, trazable, y no es ambiguo.

stakeholder. Por lo tanto, la identificación de los *stakeholders* es de suma importancia para el éxito o fracaso de un proyecto de *software*, ya que de acuerdo con Oliveros y Martínez (2015), Johnson (2018) y Becker et al., (2019) una razón por la cual un proyecto puede quedar en la categoría de incompleto o cuestionado⁵ se debe a la pobre identificación de los *stakeholders*, lo que ocasiona requisitos incompletos.

Romero et al., (2008) consideraron que todos estos problemas que se presentan en la industria de *software*, en lo que se refiere a proyectos con requisitos incompletos, son ocasionados por un aprendizaje deficiente de la IR y en particular de la ER. Ya que se supone que un estudiante de Ingeniería de *Software* debiera tener conocimientos sobre la identificación correcta y completa de los *stakeholders*, pero no es así, lo que abre una brecha entre los conocimientos adquiridos y las necesidades de la industria.

Sevilla et al., (2019) argumentan que, en el ámbito educativo, los procesos de enseñanza aprendizaje de la ER a nivel licenciatura no se encuentran actualizados, además de que brindan información insuficiente al estudiante sobre cómo identificar correctamente a los *stakeholders*. En este sentido, Incencio Piñeiro et al., (2022) afirmaron que la IS es una materia importante dentro de la formación de lo(a)s ingeniero(a)s de *software* e ingeniero(a)s de requisitos, ya que los conocimientos otorgados por esta disciplina son fundamentales al momento de ejercer su profesión. Al ser la IS una disciplina que va de la mano con la tecnología, donde ésta se encuentra en constante evolución, significa que la información que se imparte cambia constantemente, esto provoca que los educadores de la IS tengan que investigar constantemente para actualizar la información proporcionada dentro de la educación. Sin embargo, así como la información cambia constantemente, los métodos de enseñanza también deberían de cambiar ya que hoy en día los métodos de enseñanza comunes como los libros, presentaciones, etc., no son suficientes, por lo que es necesario buscar otras alternativas que motiven tanto a los docentes como a los estudiantes a una participación más interactiva dentro de la enseñanza de la IS, como los juegos serios⁶.

La investigación de Moyano et al. (2021), por ejemplo, afirma que el aprendizaje tanto de la IR como de la ER exige técnicas y enseñanzas que sean fáciles de entender para los estudiantes, y que se debe tomar en cuenta que el aprendizaje no solo debe proporcionar conocimiento teórico, sino que también conocimiento que se pueda aplicar. Por ejemplo, en el aprendizaje de la IR se debe llevar a cabo un modelo de supuestos, restricciones y representaciones que permitan al estudiante aprender a realizar el proceso de la elicitación de requisitos, pero a pesar de que se sabe lo que se tiene que hacer, la enseñanza no siempre se lleva a cabo de manera satisfactoria. La falta de este éxito en la enseñanza de la ER radica en la incertidumbre que se deriva del proceso de transferencia de información hacia el estudiante, que no siempre es consistente. A esta dificultad se le aumenta el hecho de que el estudiante no conoce ni sabe cómo llevar a cabo una correcta y completa identificación de *stakeholders*. Como punto final los autores muestran que desafortunadamente las estrategias de enseñanza tradicionales ya no son suficientes, por lo que resulta necesario implementar nuevas técnicas en los procesos de enseñanza para que de esta forma se pueda abordar la complejidad de los

⁵ Un proyecto incompleto o cuestionado se refiere a que los proyectos fueron entregados de manera tardía, que superaron el presupuesto acordado, y/o que fueron entregados con menores características y funcionalidades de las requeridas (Johnson, 2020).

⁶ Chipia (2011) argumentó que un juego serio permite desarrollar actividades que se encuentran basadas en escenarios reales, en donde se es posible asumir un rol dentro del mundo real o virtual, todo esto para obtener un aprendizaje o educación dirigida a cualquier tipo de público. Un juego serio puede abarcar cualquier tipo de género y puede o no usar tecnología. Este tipo de juegos tienen como objetivo la enseñanza, así como la transferencia de habilidades y de información.

procesos y así mejorar las prácticas y el aprendizaje de los estudiantes al momento de llevar a cabo la elicitación de requisitos, específicamente en la actividad de la identificación de los *stakeholders*.

1.3. Necesidad de resolución

Ganesh y Schnittka (2014) argumentó que el aprendizaje de la IR debía tener como objetivo guiar a los ingeniero(a)s de requisitos, dentro de la industria de *software*, para que alcanzaran el éxito en el desarrollo del *software*. Sin embargo, esta misma industria no se siente satisfecha en la actualidad con el nivel de conocimientos con el que egresan de las universidades los futuros ingeniero(a)s de requisitos. Por este motivo, el autor propuso una enseñanza basada en juegos serios como un método eficiente de enseñanza/aprendizaje. Esta eficiencia se sustenta en que los juegos serios están orientados para enseñar a las personas de una manera eficiente y fácil en comparación con los métodos tradicionales (e.g., pizarra, libros, gráficos, diapositivas, etc.).

Calderón y Ruiz (2016) afirmaron que el proceso del ciclo de vida de un *software* es considerado como un enfoque de gran relevancia que debe estar presente dentro de la formación educativa de lo(a)s ingeniero(a)s de *software*. Sin embargo, dicha enseñanza se presenta la mayoría de las veces como un enfoque teórico, dando como resultado que los futuros ingeniero(a)s de *software* no tengan la experiencia ni la educación práctica necesaria para llevar a cabo con éxito sus actividades profesionales. Por tales motivos los autores sugirieron que los juegos serios se utilizaran como herramienta de aprendizaje, ya que permiten al ingeniero(a) de *software* obtener experiencia práctica de manera segura a través de experimentar con problemas reales de la industria, probar múltiples soluciones, y adquirir conocimiento.

De acuerdo con Dalpiaz y Cooper (2018), el aprendizaje de la ER se encuentra en constante evolución para hacer frente a las exigencias de *software* cada vez más complejo y cuyo desarrollo (i.e., ciclo de vida) sea más corto que los habituales. Esta evolución continua impacta significativamente en la educación y el aprendizaje de la ER, por lo que a medida que la disciplina cambia, los métodos de enseñanza y la información también lo deben hacer. Es por esto por lo que los educadores de la ER deben de hacer lo posible para proporcionar técnicas de enseñanzas actualizadas (e.g., gamificación, aula invertida, pensamiento de diseño, etc.), ya que a pesar de que los materiales didácticos como los libros son un recurso valioso, se vuelven obsoletos con facilidad, además, las prácticas impuestas en estos materiales únicamente se limitan a entornos de aprendizaje como las aulas tradicionales. Entre estos métodos actuales destaca el uso de juegos serios como una mejora para la enseñanza de la ER, ya que proporcionan entornos atractivos e interactivos que ayudan a mejorar el conocimiento teórico y práctico.

A continuación, se describen brevemente algunos juegos serios desarrollados para mejorar la enseñanza/aprendizaje de la IS y la IR:

1. En Monsalve et al., (2010) y Monsalve et al. (2018) se afirmó que en la enseñanza de la IS se deben proporcionar conocimientos que permitan que los egresados de la licenciatura puedan desarrollar *software* de calidad. Sin embargo, también se señaló que la educación proporcionada por los métodos tradicionales de enseñanza resulta insuficiente para alcanzar este propósito. Por lo que los autores propusieron “*SimulES*”, un juego serio de enseñanza en el cual, por medio de cartas educativas y estrategias diseñadas por los estudiantes, se pretende simular el proceso de desarrollo de *software*. El objetivo de este juego fue que el estudiante construyera todos los módulos requeridos que conlleva un desarrollo de un *software*, sin errores, antes que su contrincante. Los resultados obtenidos en este juego fueron que el 50% de los estudiantes que participaron en el caso de estudio obtuvieron la nota máxima

de calificación, además de que se observó que este juego estimuló en los estudiantes la habilidad para el desarrollo de un *software*.

2. Battistella et al., (2013) propusieron “*Dealing with Difficult People*” como un juego serio cuyo objetivo era evidenciar los problemas relacionados con la gestión de equipos de trabajo en un proyecto de *software*. En este juego serio, los estudiantes participaron en la simulación de una reunión para discutir el desarrollo de un *software*, aquí, un estudiante debía asumir el guion de líder de proyecto para dirigir la reunión de principio a fin y reunir las firmas de los demás integrantes para finalizarla. Sin embargo, esto no era fácil, ya que los demás miembros del equipo debían asumir roles de personas con personalidades difíciles (e.g., que se quejen por todo, que se enojen fácilmente, inconformistas, etc.). La finalidad de esto es que tanto el estudiante que representaba al director de proyecto como el estudiante que representaba al socio, pudieran manejar correctamente la reunión, con el objetivo de que ésta terminara correctamente. Este juego buscó resolver la carencia de habilidades interpersonales dentro del aprendizaje de la IS. Los resultados de evaluar este juego serio se obtuvieron a partir del caso de estudio “*Desarrollando competencias personales y habilidades sociales en ingeniería informática mediante el uso de juegos serios*”, el cual fue presentado en Calderón et al., (2018). Dicho estudio contó con la participación de 27 estudiantes de Ingeniería en Informática, de los cuales el 90% consideró que el aprendizaje obtenido durante la interacción con el juego serio fue correcta y satisfactoria, ya que se logró percibir la importancia de la comunicación dentro del equipo de desarrollo.
3. De manera similar, Ganesh (2014) propuso el juego serio de mesa “*Hard Choices*”, cuyo objetivo fue que los estudiantes lanzaran un producto de *software* al mercado antes que sus contrincantes. Los estudiantes debían interactuar con un tablero de juego dividido en casillas, para poder avanzar entre las casillas se debían tirar unos dados que indicaban el número de casillas a avanzar. En cada casilla se debería tomar una decisión relacionada con el desarrollo de un *software*, la cual ayudaría o perjudicaría al estudiante. Estas decisiones afectaban el desempeño de su producto de *software*, brindando una experiencia real y cercana a lo que se vive dentro de las empresas. Este juego buscó resolver la falta de conocimientos prácticos proporcionados dentro de la educación de la IS, ya que permitió que los estudiantes ampliaran y reforzaran los conceptos sobre esta disciplina. Los resultados obtenidos con un caso de estudio demostraron que el aprendizaje, obtenido a través del uso de este juego, fue dos veces mayor al obtenido por medio de técnicas de enseñanza tradicional en el aula.

En cuanto a propuestas para mejorar la enseñanza de la ER, se encontraron los siguientes juegos serios:

1. “*Modela*” (Blanco, 2015) es un juego serio que se dirigió tanto a profesores como a estudiantes de la ER. Este juego serio propuso resolver la falta de educación práctica impartida dentro de la enseñanza de la ER, a través del uso de diferentes modelos y escenarios para la especificación de requisitos: como el modelo verbal y los casos de uso. Los estudiantes debían ser capaces de realizar, a través de requisitos obtenidos de un problema real, diagramas UML (i.e., diagramas de caso de uso, de clases y de secuencia) de forma correcta. Además, permitía a los estudiantes poner en práctica los conocimientos sobre cómo modelar requisitos a través de los artefactos de UML. Para su validación se llevó a cabo un caso de estudio con 32 estudiantes, los cuales se dividieron en dos grupos: experimental (los que iban a probar el juego) y de control (los que iban a seguir usando los métodos tradicionales). Los resultados obtenidos fueron que el 87% de las personas que hicieron uso del juego afirmaron haber

obtenido conocimiento nuevo, mientras que el 62% de los participantes consideró que el juego serio fue de gran ayuda en la obtención de dicho conocimiento.

2. El juego serio “*Biyubi*” (García et al., 2019) propuso un aprendizaje basado en juegos que sirviera como apoyo dentro de la enseñanza de la ER en la IR. Este juego, tiene como objetivo ayudar a los estudiantes a obtener experiencia práctica dentro de este rubro. En *Biyubi*, el estudiante deberá escoger como primer paso un avatar, para poder recorrer el escenario del juego (correspondiente a una biblioteca) en busca de *stakeholders* potenciales, los cuales le puedan brindar información necesaria para obtener una lista de deseos y necesidades y obtener de esta forma los requisitos solicitados en el juego. Para su validación se desarrolló un caso de estudio en el cual participaron 10 estudiantes de NovaUniversitas. Los estudiantes fueron divididos en dos grupos diferentes: un grupo de control y un grupo experimental, en donde el primer grupo debía llevar a cabo la ER de manera práctica dentro de la universidad (i.e., simulación real) y el segundo grupo debía de llevar a cabo la elicitación de los requisitos de manera práctica dentro del juego. De acuerdo con los resultados obtenidos, el grupo experimental (el grupo que probó el juego) fue aquel que tuvo el mejor desempeño en relación con el aprendizaje obtenido, ya que los resultados del cuestionario para valorar el enfoque del Aprendizaje Basado en Juegos (GBL, por sus siglas en inglés) demostraron que la afirmación de “el enfoque utilizado para realizar la práctica me permitió identificar y analizar lo que realmente era importante para los *stakeholders*” obtuvo una puntuación de 3.6/4, mientras que para el grupo de control se obtuvo una calificación de 2.6/4.
3. Ibrahim et al., (2019) propusieron la idea de que a través de un juego serio sería posible mejorar la comprensión de los estudiantes sobre la ER, a la vez que permitiera resolver la falta de experiencia práctica que tiene un elicitador de requisitos al contar únicamente con educación teórica sobre el tema. En este juego serio, el estudiante asumía el rol de elicitador de requisitos y tenía como misión el recopilar los requisitos proporcionados por los *stakeholders*. Se realizó un caso de estudio con 10 estudiantes, quienes tenían que interactuar y probar el juego serio, obteniendo los siguientes resultados: considerando a la elicitación de requisitos, se encontró que el 90% de los estudiantes pudo identificar correctamente al menos cinco de los nueve requisitos correctos; en cuanto a la identificación de los *stakeholders* no se tuvieron los resultados de aprendizaje deseados, ya que solamente el 20% de los estudiantes tuvo un desempeño favorable mientras que el 80% tuvo un desempeño poco favorable.
4. En la investigación de Chanchí et al., (2019) se planteó un juego serio que permitiera a los estudiantes mejorar su comprensión sobre la ER y la clasificación de los requisitos en funcionales y no funcionales. En este juego los estudiantes debían interactuar con una aplicación o interfaz sencilla de computadora en la cual los requisitos, ya sean funcionales o no funcionales, descendían de la parte superior de la pantalla y los estudiantes debían de colocar cada requisito en la canasta correcta (había una para requisitos funcionales y otra para requisitos no funcionales). Por cada requisito que fuera clasificado de manera correcta, el estudiante obtenía un punto, ayudando así a su puntuación final y a mejorar el aprendizaje de los estudiantes sobre la ER mientras que interactuaban con el juego. Para validar este juego se desarrolló un caso de estudio aplicado a través de cinco profesores del área de IR y ER de la Universidad de Cartagena, Universidad de Medellín, y Universidad del Quindío en Colombia, quienes llegaron a la conclusión de que el aprendizaje que se obtuvo se debió a que se trató de un juego serio que era sencillo de usar e intuitivo para los estudiantes.

Por todo lo anteriormente mencionado, queda patente la importancia de buscar una solución innovadora en la enseñanza/aprendizaje de la ER, que evite que siga existiendo una brecha entre lo que los estudiantes de licenciatura aprenden en las aulas y lo que la industria de *software* requiere en cuanto a conocimientos de los egresados. Por lo que se hipotetiza que un juego serio podría mejorar el proceso de enseñanza/aprendizaje en la identificación de *stakeholders* como primer paso de la elicitación de requisitos de *software*, a través de apoyos visuales, ejemplos y correcciones para que los estudiantes tengan una idea más clara y precisa de cómo llevar a cabo esta actividad de manera correcta y evitar repetir los mismos errores.

1.4. Delimitaciones de la Tesis

Esta Tesis está delimitada por los siguientes puntos:

1. El juego serio se enfocará únicamente a la identificación de los *stakeholders*.
2. Para la identificación de los *stakeholders* se hará uso de la plantilla de *Volere* (Robertson y Robertson, 2012).
3. La propuesta será evaluada mediante la realización de un caso de estudio en estudiantes de la Universidad Tecnológica de la Mixteca.

1.5. Hipótesis del trabajo

Considerando la problemática definida anteriormente, se plantea la siguiente hipótesis de investigación como punto de partida para realizar este trabajo de Tesis:

“El desarrollo y uso de un juego serio permitirá una mejora óptima en la enseñanza de la identificación de los *stakeholders*, en la elicitación de los requisitos, para estudiantes de licenciatura.”

1.6. Objetivos del trabajo

Considerando lo anterior, esta Tesis se desarrollará en el marco del siguiente objetivo general y los objetivos específicos definidos a continuación:

1.6.1. Objetivo general

“Desarrollar un juego serio para mejorar el proceso de enseñanza/aprendizaje de la identificación de stakeholders, en la elicitación de requisitos de software, en estudiantes de licenciatura”.

1.6.2. Objetivos específicos

Para alcanzar el objetivo general descrito anteriormente, será necesario cumplir con los siguientes objetivos específicos.

1. Analizar la importancia de la identificación correcta y completa de los *stakeholders* durante la elicitación de los requisitos de un proyecto de *software*.
2. Realizar un análisis exhaustivo de propuestas similares con la intención de identificar aspectos comunes que pudieran ser tomados en cuenta por la Tesis.
3. Desarrollar un juego serio que permita el aprendizaje de una correcta y completa identificación de los *stakeholders* a través del uso de la plantilla de *Volere*.

4. Evaluar los resultados de aprendizaje obtenidos al utilizar el juego serio propuesto contra los métodos convencionales.

1.7. Aproximación a la solución

La IS juega un papel importante dentro de la educación de licenciatura en México, ya que actualmente son más las instituciones universitarias que imparten esta disciplina dentro de sus planes de estudio (Vera, 2019). Sin embargo, debido a la poca o nula preparación práctica de los estudiantes de IS que egresan de las universidades, surge la necesidad de aplicar nuevos enfoques de enseñanza como los juegos serios. Este tipo de juegos tiene como objetivo mejorar la enseñanza, así como facilitar la transferencia de habilidades y de información, para propiciar así la generación de conocimiento en los estudiantes. Así pues, la aplicación de este método permite no sólo que los estudiantes jueguen y se diviertan, sino que también está diseñado para incluir actividades de aprendizaje que puedan introducir gradualmente los conceptos y guiar a los estudiantes hacia el objetivo final que es el aprendizaje. Por lo que en este trabajo de Tesis se propone el desarrollo de un juego serio como herramienta de apoyo para mejorar el proceso de enseñanza/aprendizaje de la identificación de los *stakeholders* en la elicitación de requisitos de *software* en estudiantes de licenciatura.

En este sentido, se optó por hacer uso de un juego serio debido a que presenta ventajas importantes para la enseñanza/aprendizaje (Larios et al., 2015; Morales-Trujillo y García-Mireles, 2020), como las siguientes:

- Actividad voluntaria. Por lo general, este tipo de actividad es elegida por el estudiante voluntariamente y es éste quien controla cuándo, dónde, cómo y de qué manera va a interactuar con el juego serio.
- Inmersión. Si un juego serio tiene un buen guion puede ayudar a que el estudiante se mantenga inmerso en él, lo que le permitirá adquirir correctamente el conocimiento proporcionado.
- Límites de tiempo y lugar. Es posible definir lugares o entornos similares a los de la vida real con los cuales el estudiante podrá interactuar durante un tiempo límite para realizar actividades concretas, ayudando así a mejorar su concentración.
- Establecimiento de reglas. Todo juego serio establece reglas, las cuales pueden enfocarse a escenarios de la vida real. Dichas reglas deben de orientar al estudiante a lograr las metas y los objetivos del juego serio.
- Creación de un grupo social. Los juegos serios son capaces de formar grupos sociales los cuales, en conjunto, son capaces de facilitar el intercambio de experiencias sobre cómo obtener un mejor desempeño durante la sesión de juego con el objetivo de cumplir una meta determinada o dar retroalimentaciones para resolver un problema.

En el contexto de esta Tesis se utilizará la Metodología para el Desarrollo de Juegos Educativos basados en Guiones Interactivos propuesta por de Lope et al., (2015) en consenso con la de Larios et al., (2015) (ver Figura 1.1). Esta metodología propone las siguientes fases y actividades para el desarrollo e implementación de un juego serio:

- **Prefase-1: Diseño de los retos educativos, las competencias básicas, y los objetivos educativos.** En esta fase, los docentes deberán determinar cuáles van a ser los objetivos, metas y competencias del juego.

- **Prefase-2: Diseño del tipo de juego.** Durante esta fase se determina el género, el control del avatar, plataforma, los usuarios futuros, nivel narrativo, área de aplicación, y la interactividad del juego.

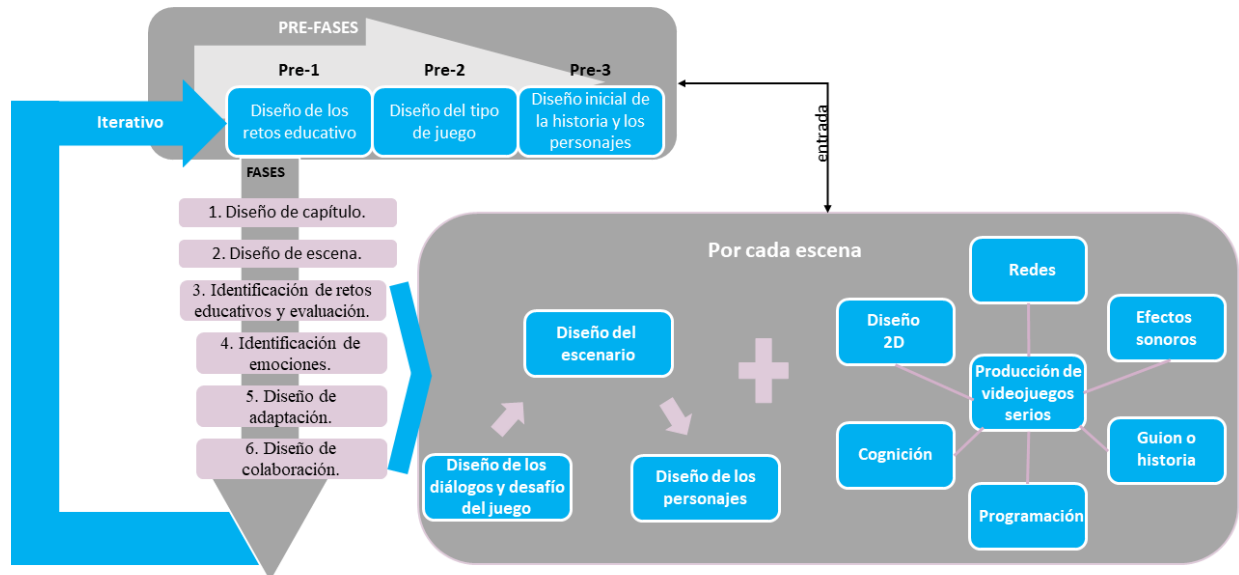


Figura 1.1. Metodología de desarrollo de juegos basada en guiones interactivos (adaptada de Larios et al., (2015) y de Lope et al., (2015))

- **Prefase-3: Diseño inicial de la historia y personajes principales.** Para que un juego esté completamente definido y para que, conforme se vaya avanzando en su dinámica, no se pierda continuidad en la historia, es necesario diseñar un argumento inicial que la sintetice. Para que esto se pueda cumplir es necesario llevar a cabo las siguientes fases:
 - *Fase 1:* Diseño del capítulo. Un capítulo se define como el elemento de más alto nivel que se utiliza para organizar la historia y facilitar la integración de contenidos. Cada juego debe tener al menos un capítulo y cada uno de los capítulos establecidos debe tener una secuencia que permita generar una historia.
 - *Fase 2:* Diseño de la escena. Cada capítulo del juego está conformado por escenas, las cuales van a determinar la trama, por lo que se deben especificar a través de un diagrama de flujo, ya que no existe un único flujo dado que existirán escenas opcionales dependiendo de la decisión del jugador. De acuerdo con Bueno (2018), cada escena debe ser descrita mediante un nombre y un resumen donde queden patentes los siguientes elementos.
 - Diseño del escenario. El diseño de un escenario es de gran importancia, ya que éste puede influir en el éxito o fracaso del videojuego. Los escenarios deben ser originales, estar bien diseñados y contar con el mayor número de detalles posibles. Mientras más características de las antes mencionadas tenga, más divertido e innovador será el juego serio.

- Diseño de los personajes. Este diseño es esencial cuando se busca crear un juego con una historia desarrollada, ya que la participación de un personaje es de suma importancia para que la historia tenga sentido.
 - Diseño de los diálogos y desafíos del juego. Los diálogos dentro de los juegos son necesarios para aportar profundidad. Regularmente éstos deben ser cortos, estar bien escritos y ser atractivos para los jugadores, ya que deben ser capaces de transmitir emociones y sensaciones para que éstos se puedan sentir involucrados con la historia.
- *Fase 3*: Identificación de retos educativos y evaluación. Es necesario que cada vez que un objetivo educativo sea alcanzado, se le haga saber al(a) jugador(a) por medio de una acción o un cuadro de diálogo.
 - *Fase 4*: Identificación de emociones. Es necesario resaltar las partes del diálogo que provocarán una reacción emocional en el jugador, de tal forma que en el momento de que esto se lleve a cabo se genere en el(la) jugador(a) algún tipo de emoción que le incentive un mayor interés por juego.
 - *Fase 5*: Diseño de adaptación. Es necesario definir atributos que personalicen el juego para así poder determinar si el juego logra acoplarse a las necesidades y características del jugador. Algunos de estos atributos son los retos educativos, el tipo de narrativa, el tipo de interacción, etc.
 - *Fase 6*: Diseño de colaboración. Se deben definir qué desafíos, dentro de las tareas, van a realizarse en grupos de jugadores.

Es así como en esta Tesis plantea el desarrollo de un juego serio para complementar el aprendizaje de la identificación de los *stakeholders* en estudiantes que cursen la materia de Ingeniería de Requisitos de la carrera de Ingeniería en Computación de la Universidad Tecnológica de la Mixteca. Por lo tanto, este juego serio tiene como objetivo fungir como una herramienta de apoyo para los profesores que impartan la materia antes mencionada, complementando así la enseñanza basada en métodos tradicionales. Esto quiere decir que el profesor seguirá impartiendo sus clases acerca de la identificación de los *stakeholders* con los métodos tradicionales y todo lo que este proceso conlleva, para que posteriormente, cuando el profesor considere que los estudiantes ya hayan adquirido todo el conocimiento teórico necesario, se lleven a cabo sesiones en donde se evalúe tanto el uso del juego serio como sus resultados. Se pretende que los estudiantes no solo lleven a cabo de forma correcta la identificación de los *stakeholders* interactuando como personajes dentro del juego serio, sino que también puedan reforzar sus conocimientos sobre la IR.

1.8. Evaluación empírica

Para que la IS se convirtiera en una verdadera disciplina fue necesario introducir métodos de evaluación encontrados en otras disciplinas, uno de éstos fue la evaluación empírica (Cruz et al., 2019). La evaluación empírica en la IS permite alcanzar una mayor validez y confiabilidad en los resultados de un proyecto de *software*. De acuerdo con Wohlin (2021) y Wohlin y Rainer (2022), en la IS existen 7 métodos importantes de evaluación empírica: casos de estudio, aplicación de entrevistas, análisis de archivos, investigación de acción, análisis del discurso, aplicación de encuestas, y trabajo con grupos de enfoque o *focus group*.

Por lo tanto, en este trabajo de Tesis se presenta la realización de un caso de estudio, ya que éste permite evaluar los beneficios de usar métodos y herramientas alternativas, y asegura de una manera efectiva que los cambios en los procesos generan los resultados deseados. Aunado a lo anterior, un caso de estudio permite que la recolección de datos se lleve a cabo durante un período de tiempo determinado (e.g., el desarrollo de un proyecto de *software*) (Chaves y Weiler, 2016).

De acuerdo con Wohlin (2021), existen tres estrategias para desarrollar un caso de estudio:

1. Comparar los resultados obtenidos de la aplicación de la propuesta contra una línea base.
2. Realizar dos proyectos gemelos (i.e., ambos proyectos deben tener las mismas características: tamaño, participantes, escolaridad, etc.) tomando uno de ellos como línea de base.
3. Aplicar la nueva propuesta a algunos componentes individuales seleccionados y comparar los resultados con los obtenidos en los componentes individuales no seleccionados.

En este trabajo de Tesis se presentan los resultados de optar por la segunda opción, dado que se realizó una evaluación empírica para determinar la efectividad del juego creado en un entorno educativo real, pero definiendo un equipo experimental (que usaría dicho juego como complemento de un curso teórico) y un equipo de control (que usaría únicamente un método tradicional de enseñanza).

1.9. Estructura preliminar de la Tesis

El contenido del presente documento de Tesis se detalla a continuación.

- En el Capítulo 2 se desarrolla el marco teórico que da sustento a los juegos serios, así mismo se describen las propuestas similares existentes para hacer una comparativa entre ellas.
- El Capítulo 3 describe el desarrollo del juego serio creado como soporte a la enseñanza/aprendizaje de la identificación de los *stakeholders*.
- En el Capítulo 4 se presentan los resultados obtenidos en el caso de estudio realizado para que estudiantes de la carrera de Ingeniería en Computación de la Universidad Tecnológica de la Mixteca utilizaran el juego serio como complemento de un curso sobre Ingeniería de Requisitos.
- El Capítulo 5 documenta las conclusiones del desarrollo de este trabajo de Tesis.
- La sección de Referencias bibliográficas concentra los estudios utilizados para sustentar la investigación producto de esta Tesis.
- Finalmente, en la sección de Anexos se presentará cualquier información adicional que se haya utilizado para facilitar el entendimiento del documento.

2. Marco teórico

2.1. La importancia de la Ingeniería de Requisitos en la industria de *software*

Cada uno de los problemas presentes dentro de la Ingeniería de *Software* es único, dando como resultado modelos y soluciones específicos para poder resolverlos. Sin embargo, a pesar de esta individualidad, estos problemas comparten un conjunto de métodos complejos enfocados a encontrar una solución. Es decir, toda solución requiere de la realización de un enfoque sistemático, riguroso y cuantificable para el desarrollo, uso y mantenimiento del producto de *software*. En el contexto de esta Tesis, una solución debería facilitar que los estudiantes y/o profesionistas apliquen la IR y sean capaces de investigar, obtener, desarrollar y capitalizar el conocimiento explícito y tácito del *software*, ya que en caso contrario, no podrán llevar a cabo la elicitación de requisitos correctamente (Serna y Serna, 2019).

Sevilla et al., (2018) argumentaron que, durante el proceso de la IS, es necesario la aplicación de la IR, ya que su implementación es imprescindible para el desarrollo exitoso de un *software*. Por lo tanto, el éxito de un proyecto de *software* se debe en gran medida al uso correcto de la IR, ya que llevar a cabo de manera efectiva la elicitación de requisitos, dará como resultado que se satisfagan los deseos y necesidades de los *stakeholders*.

De acuerdo con Franch et al. (2022), la IR es indispensable dentro de la industria de *software*, ya que ayuda a los analistas a recaudar información clave que les permitirá configurar un *software* exitoso, que sea adecuado y que resulte rentable dentro del mercado. De igual forma, Becker et al., (2019) consideran que la IR determina la calidad tanto del diseño como de la implementación del producto de *software*, por lo que se le asigna una importancia alta a esta etapa. Sin embargo, su importancia no solo radica en este aspecto, si no que este proceso bien implementado puede ayudar a determinar si un producto final de *software* es exitoso, incompleto o fracasado.

Por lo tanto, dentro de un proyecto de *software*, una “buena” IR empieza con una “buena” ER, ya que para poder construir un producto exitoso es necesario entender las necesidades y deseos de los *stakeholders*, determinar sus expectativas y tener una definición correcta del dominio del problema. Por todo esto, la IR tiene un impacto importante para el éxito o fracaso del *software*, ya que una mala aplicación de ésta puede dar como resultado un producto de baja calidad (Shafiq et al., 2018).

En este mismo sentido, Franch et al., (2022) argumentan que la IR es relevante también para los profesionales de la industria puesto que les permite replicar, a largo plazo, los resultados de cientos de investigaciones que se realizan sobre diferentes tipos de proyectos. Mejor aún, la relevancia de esta disciplina es útil también para la industria misma, puesto que la IR es actualmente reconocida como una disciplina independiente que provee una variedad de enfoques, herramientas y métodos que ayudan a la realización de un proyecto exitoso. De hecho, Méndez (2018) afirmó que la importancia

de la IR no debería someterse a discusión, ya que, al ser una disciplina interconectada con otros procesos involucrado en el desarrollo de un proyecto, cualquier error o decisión incorrecta que sea tomada en esta fase puede generar proyectos fracasados. Por consiguiente es necesario prestarle la importancia que se merece y capacitar a los ingenieros de requisitos para poder llevar a cabo esta fase correctamente, ya que los problemas que a menudo resaltan cuando se comete algún error dentro de la IR se relacionan con la incertidumbre y los factores humanos (e.g., experiencias, miedos, creencias).

La IR se encuentra presente en varias ramas del desarrollo de *software* y algunos ejemplos de éstas son las siguientes:

- Desarrollo Global de *Software* (GSD): En este tipo de proyectos la IR es considerada esencial, ya que la comunicación entre l(a)os ingenier(a)os de requisitos con lo(a)s demás *stakeholders* juega un papel fundamental. Debido a que, en el GSD, los *stakeholders* pueden estar distribuidos geográficamente por todo el mundo, es necesario aplicar correctamente la IR para que de esta forma se evite la pérdida y distorsión de la transferencia de información inherentes cuando se manejan diferentes idiomas y culturas de lo(a)s *stakeholders* (Shafiq et al., 2018).
- *Software start up*: La IR es vital para este tipo de empresas, ya que si se usan técnicas poco fiables durante la elicitación de requisitos (e.g., uso de técnicas incorrectas de elicitación de requisitos, aplicación de técnicas incorrectas de identificación de *stakeholders*) es seguro que se tendrá un proyecto fracasado. De igual forma, el uso ineficiente o básico de la IR por parte del equipo de desarrollo puede causar varios desafíos (e.g., aumento de presupuesto, desarrollo de requisitos incompletos), lo que puede provocar que el producto de *software* no sea el adecuado para su mercado objetivo (Tripathi et al., 2018).
- *Machine learning* (aprendizaje automático): Las decisiones dentro de un *software* de este tipo, deben basarse en el conocimiento o dominio del mercado actual y de las necesidades de los *stakeholders*, por lo que es crucial que la(o)s ingeniero(a)s de requisitos proporcionen estos datos desde una perspectiva más amplia ya que deben explicar el contexto del mercado y disminuir la brecha de interpretación que existe con los diferentes *stakeholders* (Vogelsang y Borg, 2019).
- Desarrollo ágil de *software*: Uno de los factores claves en este tipo de desarrollo es el uso e implementación de una IR “dinámica”, ya que permite generar de manera eficiente prácticas que involucran requisitos livianos (e.g. creación de historias de los *stakeholders* o *storybooks*, iteraciones cortas que den como resultados un *software* funcional, desarrollo de prototipos rápidos o livianos), lo que agiliza la elicitación de requisitos (Ochodek y Kopczyńska, 2018).
- Desarrollo de videojuegos: El uso de la IR dentro de la industria de los videojuegos no es muy común e incluso algunas empresas consideran que sus métodos no son aplicables al desarrollo de un videojuego. No obstante, la IR ayuda en la mejora del desarrollo y definición de los requisitos tempranos y además proporciona una visión más detallada de estos, disminuye los procesos de recopilación y análisis de datos, y proporciona métodos que ayudan a la generación de buenas ideas en el desarrollo de los videojuegos (e.g., metodologías de sistemas, diseño de aplicaciones conjuntas, construcción de requisitos dirigidos por el usuario) (Berg Marklund et al., 2019).
- Transformación digital (e.g., IoT, sistemas adaptativos): La IR también es de vital importancia para este tipo de desarrollo de *software*, ya que la(o)s ingeniera(o)s de requisitos deben de llevar a cabo tareas que son relevantes dentro de su labor cotidiana como, por ejemplo, obtener información de los sensores que se encuentren disponibles para identificar carreteras heladas, en donde es necesario evaluar con precisión los peligros reales a los cuales los transeúntes se

pueden enfrentar. Además, no solo se deben elicitar los requisitos de los *stakeholders*, sino que también se deben especificar para el desarrollo de un sistema ideal en funcionamiento (Weyer et al., 2020).

- *Crowdsourced* (colaboración colectiva): Aquí la IR cumple un papel importante para el desarrollo del *software*, ya que ésta se encarga de mejorar la calidad del mismo y de todo lo que conlleva la elicitación de requisitos, como lo son los asuntos monetarios para llevar a cabo esta actividad. Además, se encarga de la comunicación con los *stakeholders*, en donde su inclusión es importante, ya que como su nombre lo dice, se necesita un número importante de *stakeholders*. De esta forma, el equipo de desarrollo de *software* logra tener un conocimiento más actualizado que ayuda en el desarrollo exitoso del *software* (Khan et al., 2021).
- *Crossover services* (servicios cruzados): La IR es fundamental para el desarrollo de estos servicios, ya que para garantizar su éxito es necesario la elicitación de requisitos de alta calidad, así como reducir los esfuerzos dentro del desarrollo del *software*, por lo que, si no se tiene una buena IR, es probable que el producto final no tenga el rendimiento y/o desempeño esperado provocando de esta forma que sea considerado como un producto no exitoso (Liu et al., 2021).

De acuerdo con Gambo et al. (2018), en la IR existen varios factores a los cuales se les debe de dar mayor prioridad y uno de estos factores es precisamente la elicitación correcta de los requisitos de *software*, ya que ésta permite gestionar las opiniones proporcionadas por los *stakeholders*, además de que facilita la detección de incumplimientos como la falta de implementación de requisitos importantes para el desarrollo del *software*, el exceso en el presupuesto o la entrega fuera de tiempo.

Como se puede apreciar con lo anteriormente expuesto, la IR es considerada por la industria como una de las fases críticas en el desarrollo de productos de *software*, ya que una mala aplicación de esta disciplina puede provocar errores fatales, que el tiempo de entrega se extienda y que el presupuesto sobrepase al definido inicialmente. Por lo tanto, para tener más probabilidades de desarrollar un *software* exitoso, es necesario que durante la etapa de IR se definan los objetivos, se planifique correctamente el proyecto y se establezca la participación con los *stakeholders*, siendo este último punto un factor clave para el éxito del proyecto (Lima et al., 2022).

2.2. Problemas dentro de la industria

Los problemas que existen y que se presentan dentro de la IR son numerosos y variados, por lo que incluso se han desarrollado iniciativas (e.g., NaPiRE⁷) en donde se abordan las diferentes dificultades con las que se puede encontrar un(a) ingeniero(a) de requisitos al implementarla. Sin embargo, el contratiempo que existe con este tipo de iniciativas es que no son suficientes (esto se debe a que a los profesionales les es complicado compartir su información y experiencia debido al entorno competitivo que existe dentro de la industria) y además no alcanzan a abarcar todos los problemas ocasionados dentro de la IR, ya que es un entorno altamente individual (i.e., los problemas que se trabajan dentro de esta fase rara vez se resuelven de la misma forma) (Méndez et al., 2017).

⁷ La iniciativa “Nombrar el dolor en la Ingeniería de Requisitos” (NaPiRE, por sus siglas en inglés) surgió para solucionar la carencia de una base empírica general para la IR, así como las necesidades que necesitaban ser atendidas. Dicha iniciativa obtuvo datos relacionados con la IR a través de encuestas aplicadas tanto a investigadores como a profesionales del área (Méndez, 2018).

En resumen, algunos de los problemas a los que se enfrentan los profesionales de la industria de *software* al desarrollar un producto son:

- La continua evolución tecnológica.
- Los constantes cambios sociales.
- Los cambios ocasionados dentro de su mercado meta.

En consecuencia, las empresas deben prestar atención primordial a la IR ya que, con su ayuda, se podrán adaptar a estos cambios para así lograr sus objetivos y metas establecidas, y de esta forma entregar un producto exitoso. Por otro lado, si la IR no recibe la importancia adecuada se corre el riesgo de desarrollar un producto fracasado, ocasionando que las empresas pierdan un porcentaje considerable de sus inversiones debido a la baja calidad y rendimiento del producto (Durmic, 2020).

Por lo tanto, la IR debe asegurarse de que los requisitos, tanto de forma individual como en conjunto, cumplan con características de calidad (e.g., completos, factibles, rastreables), tal y como lo define el estándar ISO/IEC/IEEE 29148:2018 (ISO/IEC/IEEE, 2018). Esto permitirá el desarrollo de un producto exitoso; sin embargo, no siempre es posible alcanzar el éxito dentro de un proyecto y esto se debe a que las tareas que deben de ser implementadas por lo(a)s ingeniero(a)s de requisitos son, en su mayoría, desafiantes. Por lo que estas personas necesitan ser bastante meticuloso(a)s al momento de implementar la IR, ya que en esta etapa se requiere cubrir numerosas responsabilidades puesto que si no se elicitan correctamente los requisitos, existe el riesgo de que se realice un proyecto diferente al propuesto originalmente (Mohabuth, 2019).

Como se mencionó anteriormente, la IR se encuentra presente en distintas ramas de la industria de *software* y en cada una de ellas tiene un papel primordial para el desarrollo de un proyecto exitoso. Sin embargo, si no se aplica correctamente, ésta puede resultar siendo más perjudicial que de ayuda. A continuación se resumen algunos ejemplos de distintos problemas que se pueden presentar en las diferentes ramas del desarrollo de *software*:

- *GSD*: Una de las ventajas que tiene el GSD es que los *stakeholders* pueden estar distribuidos geográficamente por todo el mundo; sin embargo, esto a su vez representa un contratiempo importante para la elicitación de los requisitos, ya que esta distribución provoca uno de los problemas más graves y frecuentes dentro de este tipo de desarrollos: la pérdida y distorsión de la comunicación causadas por las diferencias en los idiomas de lo(a)s *stakeholders*, así como los problemas culturales (creencias y costumbres). Si esto no es tomado en cuenta, puede conducir al desarrollo de un producto de *software* no exitoso (Shafiq et al., 2018).
- *Software start up*: A pesar de que estas empresas son esenciales para la economía de un país (debido a que son proveedoras de trabajo y de productos nacionales a nivel global), la mayoría de las veces incurrir en malas prácticas de la IR (e.g., falta de comunicación e interpretación entre el(la) ingeniero(a) de requisitos y lo(a)s *stakeholders*, uso de prácticas de la IR no estandarizadas) como consecuencia de sus recursos limitados y del personal con poca o nula experiencia en el desarrollo de *software*, provocando así que el producto final de *software* no sea adecuado para su mercado objetivo y, por ende, fracase (Tripathi et al., 2018).
- *Machine learning* (aprendizaje automático): La IR es considerada como una fase difícil en este tipo de aprendizaje ya que, por ejemplo, para los Sistemas Auto Adaptativos (SAS, por sus siglas en inglés) la elicitación de requisitos resulta ser todo un desafío puesto que es intrínsecamente incompleta, debido a que depende de datos actuales, es decir, que los requisitos se adapten a las circunstancias actuales. Sin embargo, esto puede provocar también el desarrollo de un *software* no exitoso (Vogelsang y Borg, 2019).

- *Desarrollo ágil de software*: A pesar de sus diferencias con el desarrollo tradicional de un *software*, ambos enfoques (tradicional y ágil) comparten la mayoría de los problemas ocasionados dentro de la IR (e.g., interfaces cambiantes, mala elicitación de requisitos funcionales y no funcionales, ambigüedad entre los requisitos). Sin embargo, uno de los problemas importantes al que se enfrenta la IR dentro del desarrollo ágil son las necesidades cambiantes de los *stakeholders*, ya que esto puede provocar que el producto de *software* desarrollado en la iteración anterior tenga problemas de compatibilidad con la interfaz actual. De igual forma, puede provocar la generación de documentación inadecuada, así como problemas de priorización entre los requisitos (Rasheed et al., 2021).
- *Desarrollo de videojuegos*: Son varias las empresas de este sector que no aprovechan la IR y esto se debe a que no se ha encontrado la forma correcta de rastrear y gestionar los requisitos no funcionales o de establecer pautas de comunicación con lo(a)s *stakeholders* (Berg Marklund et al., 2019).
- *Crowdsourced* (colaboración colectiva): Aunque dentro de este tipo de desarrollo existe el potencial para hacer uso de la IR, no hay información suficiente para adaptarla a proyectos de este tipo, provocando así que el producto final no tenga la calidad suficiente. Esto se debe en mayor parte a que lo(a)s ingeniero(a)s de requisitos no logran identificar y/u organizar correctamente los requisitos de todos los *stakeholders* con lo(a)s que este tipo de empresas colaboran (Khan et al., 2021).
- *Crossover services* (servicios cruzados): Uno de los grandes retos a los que se enfrentan estos servicios es la elicitación de requisitos, ya que los errores dentro de esta fase pueden provocar el retrabajo ocasionando así un incremento en el tiempo de desarrollo, aumentos en el presupuesto y, por ende, la entrega de productos de baja calidad (Liu et al., 2021).

Ahora bien, un factor importante que debe considerarse para lograr el éxito dentro de la IR es que lo(a)s ingeniero(a)s de requisitos se informen sobre cuáles son los posibles problemas con los que se pueden encontrar al momento de desarrollar un *software*. En este sentido, la iniciativa NaPiRE (Méndez, 2018) identificó los 21 problemas principales dentro de la IR, así como la frecuencia con la que éstos aparecen en los proyectos de *software* (ver Figura 2.1).

En la Figura 2.1 se pueden apreciar algunos de los problemas más frecuentes dentro de la ER (primera etapa de la IR): relación débil con lo(a)s *stakeholders*, débil acceso a las necesidades de lo(a)s *stakeholders*, soporte insuficiente para lo(a)s *stakeholders* y, defectos de comunicación con lo(a)s *stakeholders*) así como la obtención de requisitos incompletos u ocultos. Por lo que es importante llevar acabo esta etapa correcta y completamente, ya que en caso contrario puede provocar que las necesidades y expectativas de lo(a)s *stakeholders* estén mal elicidadas, lo que ocasiona una pobre calidad de los requisitos (e.g., requisitos técnicamente inviables), que los presupuestos se excedan, que lo(a)s *stakeholders* no estén satisfechos con el proyecto final y que, por ende, éste no sea exitoso. Estos problemas en su mayoría son ocasionados por una mala interacción y comunicación con los *stakeholders* evitando que se puedan obtener correctamente sus deseos, necesidades y limitaciones. Por consiguiente es necesario hacer lo posible para evitar problemas en esta etapa, ya que es aquí donde los errores suelen ser críticos, costosos de eliminar y, por lo general, difíciles de resolver, ocasionando que sean arrastrados a etapas posteriores y dificultando así el desarrollo exitoso de un proyecto (Zahid et al., 2020).

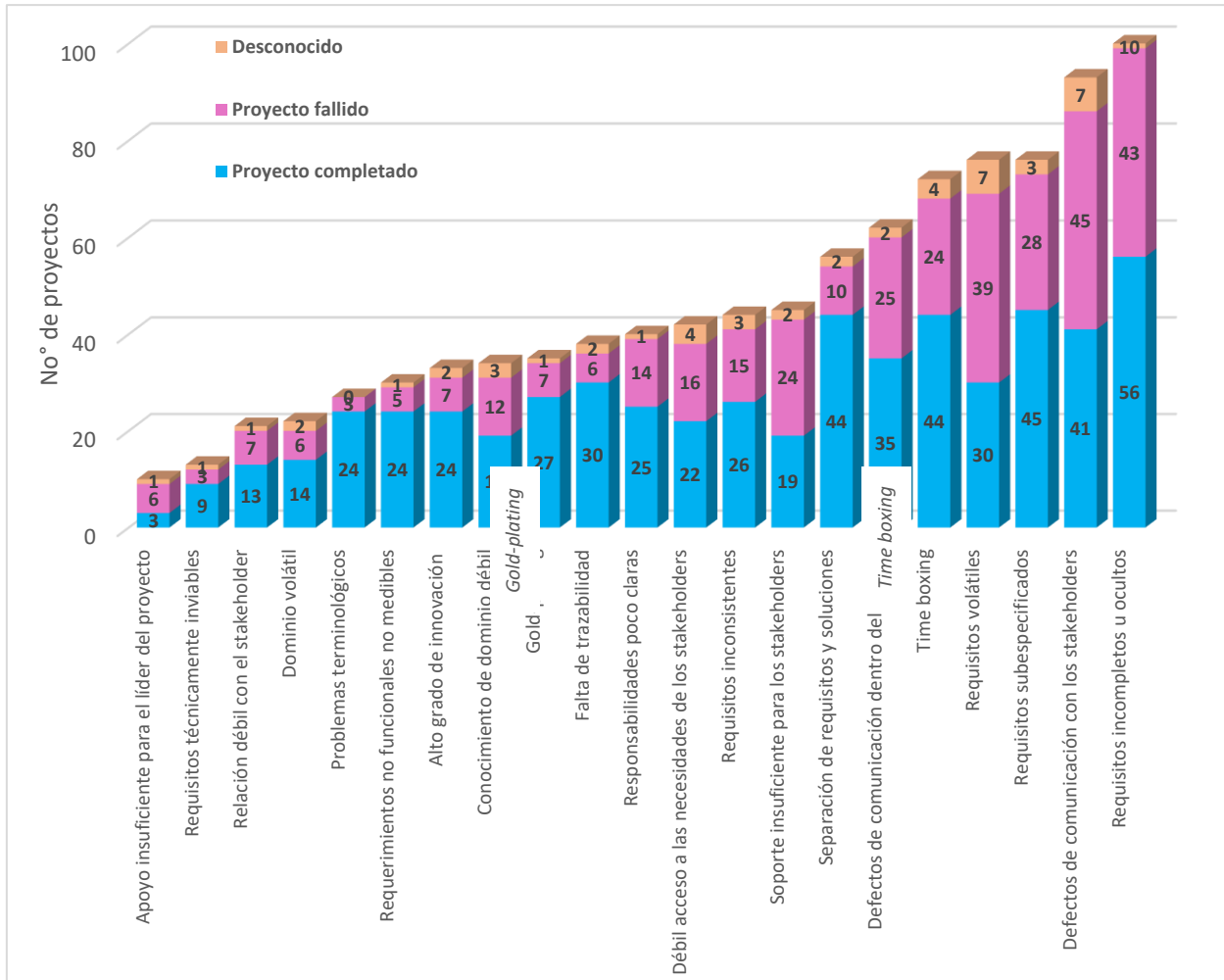


Figura 2.1. Problemas clave en la Ingeniería de Requisitos y su criticidad (tomada de Méndez (2018))

Como se ha explicado anteriormente, la ER cumple una función primordial dentro de la IR, ya que una elicitación incorrecta de los requisitos es considerada como una de las principales causas por la que un *stakeholder* no está conforme con el producto final. Otros factores importantes para tomar en cuenta dentro de la IR son lo(a)s *stakeholders*, ya que la ER requiere de una alta interacción con ello(a)s, por lo que lo(a)s ingeniero(a)s de requisitos deben de asegurar una correcta y completa comunicación, ya que una “mala” comunicación conllevará a una “mala” elicitación de requisitos que puede conducir finalmente al desarrollo inadecuado de un producto (Mohabuth, 2019).

En conclusión, es posible afirmar que si la IR se lleva a cabo de manera incorrecta existe un alto riesgo de que se termine desarrollando un proyecto fracasado.

2.3. Enseñanza de la Ingeniería de Requisitos en estudiantes de pregrado

Zainuddin et al., (2019) afirmaron que, desde una perspectiva educativa, la IR debe ser considerada como un curso obligatorio dentro de las universidades debido al impacto que tiene en la industria de *software*. En este sentido, la comúnmente llamada Educación de la Ingeniería de Requisitos (REE, por sus siglas en inglés) tiene como objetivo principal presentar el proceso de la IR al estudiante proporcionándoles herramientas para llevar a cabo correctamente la elicitación, el análisis, la especificación, la validación y gestión de los requisitos. Para lo cual es necesario que se

busque una forma en la que los educadores puedan transmitir correctamente la información de la IR a los estudiantes, teniendo en cuenta que esta materia es difícil de aprender dentro de la IS debido a su alto contenido teórico.

Dentro de la industria de *software*, el conocimiento de la IR es fundamental ya que la calidad del trabajo desarrollado por los profesionales puede verse como una consecuencia directa de la calidad de la educación sobre IS recibida en las Universidades. Lamentablemente existe una discordancia entre las habilidades que un(a) estudiante obtiene en las instituciones educativas y las que verdaderamente se necesitan y se usan dentro de la industria (Garousi et al., 2019).

La formación del estudiante universitario, en cuanto a la IR, debiera estar relacionada con el desarrollo de *software* de calidad, ya que su aplicación correcta es determinante para la calidad y éxito de un producto de *software*. Sin embargo, es considerada como una ocurrencia dentro del plan de estudios de varias Universidades que imparten la IS, ya que con frecuencia los cursos proporcionan únicamente aspectos genéricos de la IS y ejemplos poco realistas, lo que puede provocar una comprensión mínima de la IR por parte del estudiante y, por ende, egresar de la Universidad con nulo o poco conocimiento y experiencia para llevar a cabo un proyecto de *software* (Daun et al., 2023). Es por este motivo que implementar el proceso educativo de la IS e IR es una tarea importante no solo para los estudiantes, sino también para los profesionales e investigadores de estas áreas y esto se debe a que en los últimos años se ha generado la idea dentro de la industria, de que los futuros profesionales del *software* deben tener un conocimiento general y fundamental de la IS e IR (Epifânio et al., 2019).

En la actualidad, la industria de *software* ha demostrado interés en competencias específicas que deben tener lo(a)s ingeniero(a)s de requisitos (e.g., pensamiento analítico y habilidades de comunicación), esto se debe a que dichas capacidades son importantes para los puestos profesionales de desarrollo de *software*; sin embargo, muchas veces estas características no son cubiertas en su totalidad en libros y cursos sobre la IR (Hertz y Spoletini, 2018).

Es por esto por lo que la IR se debe transformar en una disciplina central de la IS dentro de la educación universitaria, ya que el tener una correcta y completa IR se traduce como el desarrollo de un *software* con requisitos funcionales y completos y, que, además, tiene más oportunidades de ser exitoso y de disminuir la probabilidad de que se sobrepase el presupuesto inicial. Para lo cual se requiere de un plan de estudios de la IR en el que se implementen, de manera equilibrada, la información teórica y la práctica (Daun et al., 2023).

Sin lugar a duda, los educadores de la IS deben de hacer lo posible por transmitir la enseñanza de la IR por medios que sean fáciles de entender para el estudiante, ya que existen métodos, como el modelo acústico (e.g., pincel, diapositivas, voz), que no dan los resultados de enseñanza/aprendizaje esperados, por lo que se deben buscar estrategias que ayuden a este proceso a ser más activo y de fácil comprensión (Epifânio et al., 2019).

Por este motivo y para que los estudiantes puedan resolver correctamente problemas relacionados con la IR, es necesario utilizar dentro de la impartición de las clases algunos medios como los mencionados a continuación:

- Casos de estudio. Son considerados como una herramienta interactiva que es efectiva para la IR, ya que proporcionan enfoques efectivos que permiten que el estudiante pueda comprender los proyectos grandes y/o difíciles de representar en un entorno escolar. Así mismo, proporcionan espacios seguros en donde los estudiantes no se tienen que preocupar en caso de cometer un error o fracasar, además de que pueden ayudar a resolver preguntas detalladas, por ejemplo: “¿cómo?” y “¿por qué?” lo que les proporcionará una visión más precisa del comportamiento del *software* (Kurkovsky et al., 2019).

- Juegos serios. Este tipo de juegos ofrecen beneficios clave en el proceso de aprendizaje de la IS e IR, ya que proporcionan un espacio seguro en donde los errores son aceptables, puesto que al final “es solo un juego”. Además, el estudiante no se siente comprometido a participar, por lo que resultan divertidos, lo que lo alienta a expresar su creatividad y emoción, factores importantes en proyectos de *software* (Kurkovsky et al., 2019).
- Aprendizaje Basado en Proyectos (PBL, por sus siglas en inglés). Al participar en proyectos de este tipo, los estudiantes pueden adquirir la habilidad de trabajar en equipo, un pensamiento crítico, mejores habilidades de comunicación, etc., además, está comprobado que este tipo de aprendizaje mejora el rendimiento de los estudiantes y mejora las habilidades y el uso de herramientas dentro de la IR (Nwokeji et al., 2018).
- Entorno basado en equipos: Este método de aprendizaje es de gran ayuda para que los estudiantes implementen la colaboración, además, desarrolla la participación, reduce los niveles de nerviosismo y vergüenza a la par de brindar aprendizaje entre pares. En este tipo de aprendizaje, se busca que se reafirmen los conceptos, las experiencias y los pensamientos (Kurkovsky et al., 2019).
- Aprendizaje situado: Este tipo de aprendizaje está basado en el enfoque de aprender-haciendo. Su objetivo es hacer que el entorno y las actividades que se realicen en clases, sean en un ambiente lo más parecido al de la industria, con lo que se busca que el conocimiento se adquiera a partir de la actividad y el contexto proporcionado (Marques et al., 2020).

A pesar de la existencia de estos medios alternativos de enseñanza, las actividades de la IR pueden llegar a ser desafiantes para los estudiantes ya que, en muchas Universidades, los profesores de la asignatura de IS ya proporcionan a los estudiantes los requisitos del *software* a desarrollar, evitando así que esto(a)s puedan aprender y experimentar por sí mismos la elicitación de requisitos. Esto provoca que tanto los profesionales recién egresados como los estudiantes tengan malas prácticas y no tomen en serio el aprendizaje de la IR, lo que hace que opinen, en muchas ocasiones, que su aprendizaje es innecesario o trivial, hasta que se enfrentan a problemas complejos dentro de la industria que podrían ser resueltos si hubiesen tenido una enseñanza completa de la IR (Kurkovsky et al., 2019).

Para Epifânio et al., (2019) uno de los principales desafíos a los que se enfrentan los educadores de la IR es la falta de interés de los estudiantes por aprender la materia, lo que muchas veces afecta directamente el entendimiento y comprensión de esta área. Lamentablemente, en la educación de la IS, los estudiantes muestran reticencia al desarrollo completo de un *software* a partir de cero (i.e., se reúsan a realizar todas las etapas que componen al ciclo de vida, desde la elicitación correcta de requisitos hasta la liberación del producto) ya que prefieren enfocarse únicamente en la programación y se limitan a desarrollar requisitos de sistemas simples que generalmente no tienen impacto dentro de la industria de desarrollo de *software*.

Por lo anteriormente expuesto, en el siguiente apartado de este trabajo de Tesis se analizará el estado del arte que guardan los juegos serios en la enseñanza de la Ingeniería de Requisitos.

2.4. Estado del arte

2.4.1. Juegos serios

Los Juegos Serios (JS) se pueden definir como juegos que no tienen como fin principal el proporcionar entretenimiento, diversión, etc., sino que buscan que las personas que los usan puedan obtener un conocimiento educativo a base de escenarios que promueven la enseñanza (Abdellatif et

al., 2018). Los JS tienen como fin ir más allá del entretenimiento y su uso puede traer consigo grandes ventajas como la obtención del aprendizaje a través de la experiencia, ya que permiten a los estudiantes y/o profesionales de la industria poder experimentar el fracaso y buscar soluciones que sean más factibles al realizar el mismo ejercicio las veces que sean convenientes (Caserman et al., 2020; Larson, 2020).

A continuación, se mencionan algunos de los sectores de la industria que hacen uso de los juegos serios:

- Ingeniería Industrial. Específicamente en la planificación de fábricas, la enseñanza basada en JS puede ayudar a los empleados que tienen nivel *senior* a cambiar su enfoque personal en cuanto a cómo realizar un trabajo dentro de la empresa, a ayudar a los trabajadores en general a superar sus limitaciones emocionales, y a proporcionarles mayores habilidades al momento de implementar herramientas, como por ejemplo la directriz VDI 5200 (procedimiento de fabricación). Un ejemplo de un JS que es utilizado dentro de este sector es “*Factory Planner*”, un juego que busca que el usuario entienda los procesos de planificación de una fábrica a través de escenarios iterativos y procesos similares a los de las industrias reales (Bilge y Severengiz, 2019).
- Salud. Los JS pueden ayudar también a lo(a)s doctore(a)s y enfermero(a)s a que tengan mayor empatía con los pacientes mayores de edad, a confrontar dificultades como las cirugías mediante el uso de simuladores, o a mejorar el cuidado de los pacientes. De igual forma los ayuda a fomentar el conocimiento experimental y práctico al rectificar el conocimiento teórico necesario para que al momento de aplicar una inyección subcutánea, por ejemplo, puedan hacerlo correctamente. Dos ejemplos de JS para este contexto son “*Geriatric*”, un juego de rol que ayuda a mejorar la atención médica de parte del personal médico al promover y mejorar la empatía hacia los pacientes de la tercera edad, y “*The Nurse and Midwife Specialty*”, el cual presenta preguntas y respuestas a través de una interfaz interactiva que busca evaluar y mejorar el aprendizaje de las técnicas de inyección que aprenden los estudiantes de medicina (Georgieva-Tsaneva, 2019).
- Mecánica Automotriz. La enseñanza basada en JS puede ayudar a los estudiantes a fabricar componentes de manera más precisa, a entender los procesos de fabricación desde el principio al final, e incluso a obtener planos y planes de fabricación. Además, este tipo de juegos puede asesorar a los estudiantes al momento de hacer uso de máquinas pesadas que son necesarias dentro de las empresas de la mecánica (e.g. uso de fresadora). Un ejemplo de un JS que es utilizado dentro de este sector es “*Workshop*”, un taller virtual en donde el estudiante deberá realizar los mismos procedimientos que realizaría en un entorno real con el fin de simular el desarrollo de un componente (Visvizi y Lytras, 2019).
- Capacitación de pilotos. En este sector los JS se pueden presentar como simuladores virtuales, específicamente como simuladores de vehículos operados a distancia (ROV, por sus siglas en inglés), donde los profesionales que acaban de ingresar a este sector y que no tienen la experiencia suficiente, tienen en este tipo de simuladores una opción más segura y factible de aprender su manejo sin que esta actividad sea tediosa y cansada. Un ejemplo de un JS que es utilizado dentro de este sector es “*Unity3D Serious Game Engine for High Fidelity Virtual Reality Training of Remotely-Operated Vehicle Pilot*”, el cual pretende ayudar a los futuros pilotos a operar correctamente un submarino a través de prácticas de realidad y alta fidelidad que se desarrollan dentro de una simulación virtual (Chin et al., 2018).

Ahora bien, en lo que se refiere a JS enfocados en brindar soporte a áreas de las Ciencias de la Computación, destacan las siguientes:

- **Ciberseguridad.** Los JS se han posicionado como un enfoque relevante que puede complementar la educación debido a que brindan un ambiente en el que los estudiantes pueden sentirse cómodos, facilitando así el aprendizaje de la teoría sobre conceptos de seguridad y ponerlos en práctica. Por lo general, los JS en ciberseguridad permiten que los estudiantes puedan atacar y explorar vulnerabilidades en un entorno dinámico, que adquieran la habilidad para reaccionar ante estos ataques, y que puedan contratacar y defenderse, dando como resultado un aprendizaje dinámico y efectivo. Un ejemplo de un JS utilizado en el aprendizaje de esta área es “*SWPAG*”, el cual ofrece a los estudiantes la preparación suficiente para lanzar una defensa para capturar una bandera⁸ (CTF, por sus siglas en inglés) a través de una interfaz fácil de usar (Trickel et al., 2017).
- **Internet de las cosas.** Al combinar esta área con los JS se crea lo que se conoce como Juegos Serios Inteligentes (JSI). Los JSI son conocidos como tecnologías inteligentes y su principal objetivo es ser una herramienta que ayude a resolver problemas graves dentro del desarrollo de un proyecto. Un ejemplo es “*Solicitud de Participación Estudiantil*” (SEA, por sus siglas en inglés), un juego que le proporciona al estudiante las herramientas necesarias para entender un marco modular de trabajo, el cual va desde la detección de datos hasta su aplicación (Henry et al., 2018)
- **DGS.** Los JS ayudan a los estudiantes a adquirir experiencia para ser buenos líderes de proyectos, además, por medio de ejercicios interactivos, permiten que éstos puedan ser capaces de tomar decisiones correctas que conlleven al desarrollo exitoso de un *software*. Los JS también proporcionan el aprendizaje suficiente para que los estudiantes sean capaces de entender la importancia de la comunicación con los *stakeholders* y el por qué estos son importantes en el desarrollo exitoso de un *software*. Un ejemplo de un JS utilizado dentro de esta área es “*GSD-AWARE*”, en el cual el estudiante desempeña el rol de líder del proyecto y debe tomar las decisiones necesarias que conduzcan al desarrollo de un *software* exitoso (Vizcaíno et al., 2019).
- **IS.** Los JS desarrollados para esta área permiten proporcionar experiencias educativas divertidas e interesantes y, dependiendo del enfoque que le den al juego, éstos pueden proporcionar a los estudiantes los pasos adecuados para poder gestionar correctamente una empresa de *software*. En este sentido, es posible encontrar JS en diferentes presentaciones (e.g. juegos de mesa, simuladores, juegos de computadora) y esto se debe a su importancia y facilidad de uso al momento de proporcionar aprendizaje. Un ejemplo de un JS de este tipo es “*Software Inc.*”, donde a través de interfaces iterativas, en las que el estudiante tiene que crear su propia empresa desde cero, se le va guiando por las diferentes etapas de la gestión con el fin de proporcionarle la información suficiente para que la aplique correctamente a su propia empresa (Caeiro-Rodríguez et al., 2021).

En el ámbito educativo, la relación que existe entre el uso de los JS y el aprendizaje ha sido positiva, dando buenos resultados en lo que respecta a la experiencia y el aprendizaje de los estudiantes. Por tales motivos, los JS han logrado la aceptación dentro del sector educativo, ya que, a

⁸ La competencia de seguridad informática “consiste en un conjunto de acertijos (o desafíos) de seguridad informática que involucran ingeniería inversa, corrupción de memoria, criptografía, tecnologías *web* y más. Cuando los jugadores los resuelven obtienen una “bandera”, una cadena secreta que se puede canjear por puntos. Cuantos más puntos gane un equipo, más alto se moverá en la clasificación” (Ošlejšek et al., 2019).

través de su uso correcto, se proporcionan medios interactivos para que los estudiantes se concentren y así puedan lograr los objetivos planteados por los educadores.

En este tipo de entornos, particularmente en los que están enfocados a estudiantes de pregrado, tanto la temática como el contenido de los JS deben ser los adecuados para atender correctamente las necesidades educativas con el fin de garantizar una educación exitosa (Zhonggen, 2019). Además de las características anteriormente mencionadas, los JS deben de seguir ciertas peculiaridades que los hagan destacar dentro del ámbito educativo y algunas de éstas son las siguientes:

1. **Objetivos claros.** Los JS deben crearse de tal manera que los objetivos educativos establecidos en un inicio sean fáciles de entender. También es necesario garantizar que los estudiantes siempre tengan conocimiento de cómo llevar a cabo una acción y, en caso de que las actividades o acciones no sean fáciles de intuir, se tendrá que incluir un tutorial que deberá ser puesto al inicio del JS (Caserman et al., 2020).
2. **Contenido interactivo:** Este tipo de contenido promueve el mejoramiento de la concentración de los estudiantes al involucrarlos activamente en el juego, enriqueciendo así, el proceso de aprendizaje (Anastasiadis et al., 2018).
3. **Comentarios apropiados sobre el progreso.** Los JS deben de incluir realimentaciones e información con las cuales se pueda evaluar el avance del estudiante con respecto al aprendizaje obtenido. Es recomendable proporcionar información no solo al finalizar la sesión de juego, sino también durante la misma, ya que ésta representa una ayuda para que el estudiante no cometa los mismos errores durante sesiones posteriores (Caserman et al., 2020).
4. **Promoción del sentimiento de motivación.** Dicho sentimiento permite que los estudiantes desarrollen mayor interés por el tema que están aprendiendo. Además, el aprendizaje obtenido puede ayudarlos a obtener mejores logros académicos y desarrollar un comportamiento crítico y cooperativo, además de que, cuando el JS involucra la interacción de más de un estudiante, mejorar sus habilidades sociales (Anastasiadis et al., 2018).
5. **Correctitud en el contenido del dominio.** Se debe ser cuidadoso con el contenido plasmado en los JS, puesto que, de lo contrario, se podría afectar considerablemente el aprendizaje de los estudiantes. Es decir, la información proporcionada en los JS no solamente debe ser correcta, sino que debe de ser presentada con un lenguaje técnico que sea adecuado tanto para el nivel cognitivo como académico de los estudiantes (Caserman et al., 2020).

Recientemente, el uso de los JS en el sector educativo se ha vuelto bastante popular debido a los beneficios que éstos aportan, por ejemplo:

- Pueden ayudar a los estudiantes a mejorar su estado de ánimo, lo que influye en que tengan mayor éxito al momento de obtener la información y, por ende, en la obtención del conocimiento (Zhonggen, 2019).
- Permiten que los educadores transmitan mejor el conocimiento y que los estudiantes tomen la iniciativa de descubrir conocimiento, al inspirarlos en aprender (Londoño Vásquez y Rojas López, 2020).
- Ayudan a mejorar el pensamiento crítico del estudiante, así como la toma de decisiones y la resolución de conflictos (Anastasiadis et al., 2018).
- Los estudiantes que hacen uso de los JS para el aprendizaje tienden a obtener mejores resultados durante sus exámenes, que los estudiantes que no hicieron uso de estos (Zhonggen, 2019).

2.4.2. La aplicación de los juegos serios dentro de la Ingeniería de *Software*

La IS es una disciplina que busca el desarrollo de *software* funcional y de calidad a través de la aplicación de métodos, técnicas, herramientas, etc. Por lo tanto, es necesario que tanto los estudiantes como los profesionales de esta área puedan combinar completa y correctamente sus habilidades técnicas y sociales con el fin de integrarse correctamente a la industria y esto se puede lograr con una correcta educación de la IS (Gómez-Álvarez y Zapata Jaramillo, 2022).

Sin embargo, la transferencia de información dentro de la enseñanza/aprendizaje de la IS es un reto tanto para los profesores como para los estudiantes. Actualmente, es cada vez más común ver que los estudiantes egresados de las carreras de IS no tienen las habilidades y conocimientos (técnicos y sociales) necesarios para trabajar en la industria, dando como resultado que el egresado(a) no pueda obtener un empleo relacionado con lo que estudió, además, se agrega el hecho de que en la mayoría de los casos, lo(a)s egresado(a)s no está familiarizados con los modelos de procesos (e.g., NMX-I-059/02-NYCE-2016 (NYCE, 2016), CMMI-DEV v2.0 (2018), etc.) ni con los estándares de *software* (e.g., ISO/IEC/IEEE 15939:2017, ISO/IEC/IEE 29148:2018, ISO/IEC/IEEE 26515:2018), o con iniciativas como SWEBOK (Bourque y Fairley, 2014) o IREB (Glinz et al., 2020), lo que minimiza su participación en actividades estratégicas dentro de la industria. Es por esto por lo que es necesario brindarle la importancia que se merecen los diferentes métodos de enseñanza que existen dentro de la IS, de tal forma que los estudiantes puedan adquirir la información y educación suficiente para superar estas deficiencias. Por tal motivo, el uso de los JS toma un papel de relevancia dentro de la IS, ya que, en la actualidad, son un recurso importante dentro de esta área para fomentar un aprendizaje más activo, fácil y dinámico (García et al., 2020).

En los últimos años, la necesidad de hacer uso de los JS dentro del área de la IS ha ido aumentando progresivamente, surgiendo como una oportunidad para mejorar los métodos de enseñanza dentro de las instituciones y para proporcionar un enfoque positivo con el cual compartir la información a los estudiantes de manera exitosa. Los JS son considerados como una herramienta que puede brindar educación de manera entretenida, a la par de facilitar los recursos para producir actividades y prácticas tanto teóricas como técnicas. Una ventaja de estas herramientas es que no es necesario realizar un JS extravagante para asegurarse de transmitir la información y la educación correctamente, sino que un JS simple puede lograr transmitir la información suficiente y de manera efectiva, proporcionando a la vez una experiencia emotiva para el estudiante (Calderón et al., 2019).

Por otro lado, a pesar de que no tiene mucho tiempo de que los JS ganaron popularidad dentro de la IS, su uso dentro de la educación de esta disciplina no es algo novedoso, ya que existen trabajos de investigación desde el año 1974, con un aumento significativo después de los años 2000. Esta forma de aprendizaje surge con la intención de reducir los desafíos encontrados durante la transferencia de educación a los futuros profesionales de la IS, ya que la educación relacionada con esta área no solo implica la programación, sino que también es necesario prestar atención a los requisitos, la gestión de un proyecto, la medición, la calidad del proceso y producto, etc. (Rodríguez et al., 2018). De acuerdo con Ibrahim et al. (2019), el uso de los JS en la IS es efectivo porque ayuda a los estudiantes a incrementar la concentración y la comprensión de los temas que son más teóricos y difíciles de explicar que en los formatos de educación tradicionales.

Al hacer uso de los JS dentro de la IS es importante llevar a cabo un enfoque específico que permita tener éxito en la enseñanza/aprendizaje, ya que los elementos que componen a los juegos son diversos. Sin embargo, cuando se habla de los JS enfocados a la enseñanza de la IS, es necesario definir su estructura. De acuerdo con Zhonggen (2019), los JS enfocados a la educación de la IS se deben regir por siete factores:

- Historia de fondo y la producción. Se refiere a la información y contexto sobre dónde y cómo se va a llevar a cabo el JS.
- Realismo. El aprendizaje obtenido al finalizar el JS debe ser el suficiente para cumplir con las expectativas de los estudiantes y educadores.
- Fácil interacción estudiante-juego. En caso de que se trate de un videojuego, el estudiante deberá ser capaz de interactuar por sí solo dentro de éste con la intención de que le sea lo más interesante posible y se obtenga su atención y compromiso en el aprendizaje.
- Adaptabilidad. El juego debe ser capaz de adaptarse a los objetivos individuales de cada estudiante.
- Comunicación. Debe existir una comunicación continua entre usuario-juego y usuario-educador.
- Realimentación. Al finalizar la interacción con el JS, el estudiante debe recibir sus resultados (puntaje final) como parte del juego, así como las observaciones hechas por el educador (comentarios, calificación, etc.).
- Sorpresas. Este factor no es obligatorio; sin embargo, se considera de importancia ya que puede influir en la efectividad del JS. Debido a que las sorpresas estimulan las estructuras cognitivas de los estudiantes también permiten que éstos retengan mejor la información. De esta forma, los JS que transmiten sorpresas durante su uso pueden ayudar al desarrollo del razonamiento y mejorar el proceso de aprendizaje.

Estos factores no solamente deben ser considerados en la creación de JS enfocados en complementar la educación sobre las diferentes fases de la IS, sino también en todos los juegos que se pretendan crear para brindar un soporte educativo en la enseñanza de otros aspectos de la IS, como por ejemplo:

- Legibilidad de código: La legibilidad es un problema importante para los desarrolladores de *software*; sin embargo, son pocas las formas efectivas que existen dentro de la educación que destaquen correctamente su relevancia. En este sentido, los JS pueden ayudar a resolver este problema al proporcionar información al estudiante sobre cómo realizarla correctamente. Por ejemplo, “*GamiCRS*” (Mi et al., 2018) es un JS que busca mejorar la enseñanza sobre la legibilidad de código a través de un mecanismo sistemático de incentivos, en donde cada logro del estudiante es recompensado con insignias que lo incentivan a aprender.
- Sistemas *blockchain*: Al ser considerados como tecnología emergente, son pocas las investigaciones que muestran una técnica y/o herramienta para la enseñanza-aprendizaje del desarrollo de este tipo de sistemas dentro de la educación de la IS. No obstante, se han desarrollado JS que ayudan a los estudiantes a entender, a través de escenarios atractivos, recompensas y puntos, lo que es un *blockchain*. Un ejemplo es “*On the understanding of gamification in blockchain systems*” (Parizi y Dehghantanha, 2018), un JS que utiliza interfaces gráficas e interactivas para que el estudiante comprenda las funcionalidades de los sistemas *blockchain* mediante la realización de actividades básicas que involucran a la minería de datos.
- Investigación sobre la IS: A diferencia de los cursos prácticos sobre la IS, la investigación sobre esta área involucra la lectura de artículos y libros técnicos que, a menudo, también resulta una tarea aburrida y tediosa para los estudiantes. No obstante, los JS pueden ayudar a mejorar

el interés y el compromiso que tienen los estudiantes con respecto a la investigación dentro de la IS. Por ejemplo, “GoRaD” (Nowostawski et al., 2018) es un JS que fue creado con el objetivo de involucrar a los estudiantes en la investigación sobre la IS a través de lecturas interactivas que son realizadas a través de la resolución de retos y la realización de actividades que los divierten mientras aprenden.

- *E-learning* (aprendizaje electrónico): La transferencia de información entre los educadores y los estudiantes no siempre se lleva a cabo de forma exitosa. En el contexto de la enseñanza a distancia de algún lenguaje de programación, por ejemplo, este problema se hace más evidente debido principalmente a la falta de sistemas de *e-learning* que procesen y proporcionen fácilmente información en línea. Por lo que últimamente se han comenzado a diseñar los JS que aborden esta situación, con el objetivo de influir de manera positiva en el aprendizaje, motivación y compromiso de los estudiantes. Un ejemplo es “CoMa” (Gasca-Hurtado et al., 2019), un JS que busca mejorar el proceso de aprendizaje de la programación del lenguaje Java a través de enfoques interactivos basados en información y figuras.
- Mejora de Procesos de *Software* (SPI, por sus siglas en inglés): En la actualidad, la SPI se ha convertido en uno de los pilares de la calidad y la fiabilidad del *software*. Sin embargo, a pesar de su importancia, es un campo poco explorado en la academia puesto que no existen muchos métodos educativos que ayuden al estudiante a comprender la relevancia de esta área de la IS en la industria. En este sentido, la propuesta denominada “*Gamification for software process improvement: A practical approach*” (Herranz et al., 2019) buscó aumentar la motivación por el aprendizaje de la SPI entre estudiantes y profesionales de la industria de *software*.
- Desarrollo ágil de *software*: En el desarrollo ágil de *software* el uso de la metodología Scrum es amplio. De hecho, es común ver que la enseñanza de ésta metodología sea parte de la mayoría de los planes de estudio de las universidades. Sin embargo, aprender esta metodología no siempre resulta fácil, ya que su entendimiento necesita el asimilar tanto conocimiento teórico como práctico, lo que a menudo resulta tedioso para los estudiantes. Así, una alternativa educativa fue propuesta en “*Relax, it’s a game: Utilising gamification in learning Agile Scrum software development*” (Naik y Jenkins, 2019), la cual se ha utilizado para dos cursos de licenciatura durante los últimos cuatro años, demostrando una mejora constante en el aprendizaje y evaluación de *Agile Scrum*, en comparación con el enfoque de enseñanza tradicional.
- Conocimiento sobre arquitectura de *software* (AK, por sus siglas en inglés): El AK es fundamental en el desarrollo del *software*; sin embargo, su implementación puede resultar tediosa incluso para los practicantes de la industria. Por lo tanto, es necesario buscar alternativas que motiven tanto a los estudiantes como a los profesionales, al mismo tiempo que aumentan su interés en aprender y aplicar los fundamentos para diseñar una arquitectura efectiva para el *software*. En este sentido, en “*The effect of gamification on software architecture knowledge management: A student experiment and focus group study*” (Mayer y Weinreich, 2019) se presenta una propuesta que le enseña a estudiantes y profesionales cómo gestionar sus perfiles de arquitectura, los cuales están conformados por requisitos, conceptos, patrones de arquitectura, calidad, estilos, etc.
- Base de datos y SQL: El Lenguaje de Consulta Estructurada (SQL, por sus siglas en inglés) es un lenguaje de programación diseñado para el uso general de bases de datos dentro de la IS. Este lenguaje en ocasiones puede llegar a ser complicado de entender debido a su complejidad y a los diferentes procesos con los que se relaciona, por lo que es necesario buscar métodos o

herramientas alternativas que ayuden a los estudiantes a sentirse motivados por aprenderlo. Un ejemplo es “*QueryCompetition*” (Morales-Trujillo y García-Mireles, 2020), un JS que consiste en una aplicación *web* que permite a los estudiantes practicar sus habilidades de SQL a través de desafíos y competencias, con el fin de reforzar sus conocimientos sobre este lenguaje.

Como se puede observar en los ejemplos anteriores, se ha prestado especial atención en la creación de propuestas educativas enfocadas a brindar soporte en la enseñanza de la IS entre las que destacan, principalmente, dos conceptos: los JS y la gamificación. A pesar de que a menudo estos términos son utilizados de manera indistinta, su significado es un tanto diferente. De acuerdo con de Carvalho y Coelho (2022), la gamificación consiste en aplicar diferentes elementos de los juegos (e.g., puntos, medallas, *rankings*, logros, recompensas) a ámbitos reales como las empresas, la educación, el deporte, etc., para motivar a los aprendices en la realización de tareas que a priori pudieran haber parecido poco atractivas con el objetivo de mejorar, por ejemplo, su productividad.

Por otro lado, ya se ha mencionado que los JS pueden presentar escenarios virtuales que facilitan el aprendizaje de habilidades aplicables en la resolución de problemas de la vida real. De acuerdo con Dalpiaz y Cooper (2018), los JS son una herramienta educativa útil para la enseñanza de la IR, ya que ayudan a los estudiantes a mejorar su comprensión sobre ésta, son entretenidos, incentivan la participación en clase, y generan un aprendizaje activo e interactivo.

Con el objetivo de establecer un panorama actual sobre el uso de la gamificación y los JS en un contexto educativo de la IR y sus diferentes procesos, en las siguientes secciones se presentan algunas propuestas actuales. Es importante mencionar que se hace especial énfasis en la fase de ER y la actividad de identificación de los *stakeholders*, objetivo fundamental de esta Tesis.

2.4.3. El uso de la gamificación en la enseñanza/aprendizaje de la Ingeniería de Requisitos

Khade et al., (2022) afirman que una de las formas “jóvenes” y emocionantes de involucrar a los estudiantes o profesionales en la obtención de requisitos de un *software* es precisamente la gamificación, puesto que ésta ayuda a incrementar su participación en la tarea. A continuación se analizan dos herramientas que se enfocan en este fin. Este análisis expone información para tres apartados: el *objetivo* de la herramienta, una breve *descripción* de sus principales funcionalidades, y *resultados* obtenidos con su evaluación empírica en un entorno real.

2.4.3.1. DMGame+: Explorando el conocimiento sobre la Ingeniería de Requisitos con gamificación

2.4.3.1.1. Objetivo

Con la creación y uso de *DMGame+*, Perini et al., (2018) argumentaron que el uso correcto de la gamificación en la educación de la IR es capaz de generar resultados favorables, ya que ésta puede proporcionar beneficios importantes a la enseñanza de los procesos, herramientas y técnicas relacionadas con dicha área de la IS. Por tal motivo, los autores buscaron crear un proceso estructurado que ayudara tanto a estudiantes como profesionales en el entendimiento y la realización de las actividades de la ER y, de esta forma, mejorar la adquisición de conocimiento en el área.

2.4.3.1.2. Descripción

DMGame+ es un marco de diseño que tuvo como objetivo principal el fusionar la Metodología de Modelado Semi-formal Orientada a Objetivos (SGOMM, por sus siglas en inglés) con el Diseño Centrado en el Usuario (UCD, por sus siglas en inglés) para crear un soporte educativo que abarcará desde la elicitación de requisitos hasta la validación de los *stakeholders*. *DMGame+* se conforma de dos partes:

1. Caso de uso ilustrativo: *DMGame+* se fusiona con *DMGame*, el cual está basado en el Proceso de Jerarquía Analítica (AHP, por sus siglas en inglés) y se encarga de realizar el proceso de priorización de requisitos colaborativos. *DMGame+* y *DMGame* proporcionan una herramienta basada en gamificación, la cual contiene elementos de juegos que se basan en puntajes (ver parte superior de la Figura 2.2), con el objetivo de recompensar a los aprendices cuando realicen una acción correcta e incentivarlos a continuar aprendiendo.

The screenshot shows the user interface of *DMGame+*. At the top, there is a navigation bar with 'Home', 'Dashboard', and 'SUPERSEDE Decision Making App'. The user is logged in as 'test_game_int test_game_int' and has 4 notifications. Below this is a summary table:

Votes	Accomplishment Percentage	Position in voting	Status	Agreement Index	Total Points
5	25 %	1	Not complete	1	
+5pt	+2pt	+5pt	-20pt	+20pt	12

Below the table is a 'Filter by criteria:' dropdown menu. The main part of the interface is a voting table with two tabs: 'Available actions' and 'Finished actions'. The table has columns for 'Comparison Criteria', 'First Requirement', 'Select your vote', and 'Second Requirement'. Each row represents a comparison criterion with a 'Vote' button. The 'Select your vote' column shows a scale from 1 to 9 with radio buttons. At the bottom, there is a pagination control: 'Go to page: 1 Show rows: 10 1-10 of 15'.

Figura 2.2. Interfaz de usuario de *DMGame+* (tomada de Perini et al., (2018))

2. Mejoramiento del diseño de *DMGame* desde una perspectiva enfocada en la IR: *DMGame+* involucra el desarrollo de cuatro fases principales que implementan y combinan SGOMM y UCD. Así, con la propuesta se proporcionan herramientas que mejoran la interacción de los aprendices en cada una de estas fases (como la experiencia, expectativas, comportamiento, etc.) y, además, permiten la resolución de tareas de manera más fácil y sencilla. Por lo tanto, para interactuar correctamente con *DMGame+*, los aprendices deben realizar las siguientes cuatro fases:
 - Fase 1: Exploración temprana. La implementación de *DMGame+* se debe realizar con grupos pequeños de aprendices donde, como primer paso, se asignan los roles que cada uno tendrá que desempeñar (e.g., líder de proyecto, analista, desarrolladores de *software* y *stakeholders*). Una vez que los roles han sido asignados, los aprendices deberán colaborar en el desarrollo del *software* de acuerdo con una descripción proporcionada por los profesores, con el fin de mejorar los objetivos de la empresa que se les asignó (i.e., dichos objetivos deberán proporcionarse junto con una descripción del proyecto de *software* asignado, ya que esto variarán dependiendo del proyecto). En esta fase se deberán elicitar los requisitos y obtener modelos que los representen.

- Fase 2: Identificación del problema. En esta fase los aprendices deberán realizar el análisis de aspectos críticos en el desarrollo de *software*, para así resaltar los principales problemas encontrados. *DMGame+* establece un tiempo limitado para llevar a cabo dicha actividad para posteriormente introducir el resultado en la herramienta.
- Fase 3: Visualización. En esta fase los aprendices deberán proponer y modelar soluciones para los problemas encontrados en la fase anterior, con el fin de lograr los objetivos establecidos en la fase inicial. Además, deberán implementar dichas soluciones para resolver los problemas encontrados. Para esto, los miembros del equipo deberán llevar a cabo un *focus group* que les permita discutir diferentes propuestas que serán implementadas en *DMGame+* dentro de un tiempo límite. Esto tiene como objetivo hacer esta tarea más desafiante e interesante para los aprendices, puesto que cada vez que logren llevar a cabo las tareas satisfactoriamente dentro del tiempo establecido, recibirán puntos y recompensas.
- Fase 4: Por último, los aprendices deberán evaluar los requisitos resultantes al usar *DMGame+*, así como la herramienta en sí.

2.4.3.1.3. Resultados obtenidos

DMGame+ fue utilizada en dos casos de estudio que se realizaron con profesionales de la industria con la aplicación del Pensamiento de Diseño y el método *Agon* (DTA, por sus siglas en inglés). El análisis de los resultados se basó tanto en la aplicación de un cuestionario después de haber terminado una sesión de trabajo con *DMGame+*, como en la resolución de un examen práctico. En general, todos los participantes coincidieron en que las necesidades de los usuarios y las estrategias psicológicas propuestas por DTA, y obtenidas considerando los principios de la gamificación, eran claras. Además, en cuanto a la adecuación de las necesidades y estrategias psicológicas para el *software* y los usuarios objetivo, los participantes coincidieron en que la adecuación alcanzada era correcta. También con respecto a la idoneidad de los elementos de *gamificación* para el *software* y los usuarios objetivo, todos los participantes estuvieron de acuerdo en que se logró hacer un trabajo excepcional. De hecho, uno de los participantes estuvo “muy de acuerdo” en que los elementos de gamificación propuestos por DTA eran apropiados para los usuarios objetivo. Considerando lo anterior, la claridad y adecuación de los conceptos propuestos por DTA a través de *DMGame+* fueron consideradas como elementos positivos como resultado de la evaluación.

2.4.3.2. Agon: Un marco de trabajo basado en la gamificación para la Ingeniería de Requisitos

2.4.3.2.1. Objetivo

Piras (2018) afirmó que el análisis de los requisitos de aceptación, así como su implementación, se ha convertido en un factor que influye en el desarrollo exitoso de un *software*. Sin embargo, la implementación de estos requisitos requiere de mucha experiencia por parte de lo(a)s ingeniero(a)s de requisitos provocando que el proceso de diseño de *software* sea más complejo, lento y propenso a errores. Por lo tanto, para analizar e implementar correctamente este tipo de requisitos, es necesario realizar actividades que sean respaldadas por herramientas, las cuales deben ser capaces de facilitar la comprensión de este proceso complejo. Por tal motivo, se creó la denominada *Agon-Tool*, una herramienta basada en gamificación que tiene como objetivo el permitir al ingeniero(a) de requisitos realizar la elicitación de los requisitos de aceptación, así como su análisis sistemático a través del marco *Agon*.

2.4.3.2.2. Descripción

Agon-Tool consiste en una aplicación *web* interactiva que establece las siguientes siete fases:

1. Fase 1: Identificación de requisitos básicos del sistema. En esta fase se asignan, como primer paso, los roles que cada uno de los participantes tendrá que desempeñar (i.e., analista o *stakeholder*). Una vez establecidos los roles, el analista deberá diseñar el metamodelo del *software* a desarrollar arrastrando y soltando los elementos que desee incorporar a su área de trabajo (ver Figura 2.3).

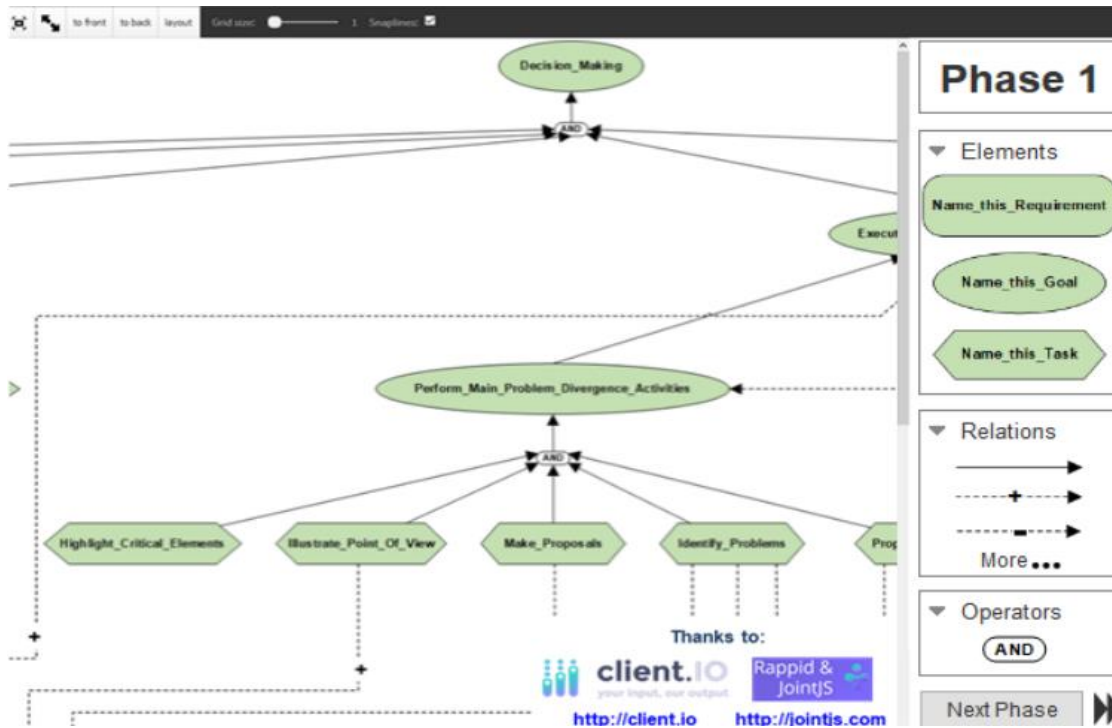


Figura 2.3. Obtención de un metamodelo de requisitos con *Agon-Tool* (tomada de Piras (2018))

2. Fase 2: Elicitación y análisis de los requisitos de aceptación. Los *stakeholders*, con ayuda del analista, deberán revisar el metamodelo diseñado anteriormente y una vez que se aseguren que es correcto, el analista deberá elicitar los requisitos de aceptación con ayuda de los *stakeholders* para introducirlos en *Agon-Tool* y realizar así la fase de análisis (ver Figuras 2.4 y 2.5).
3. Fase 3: Caracterización del contexto. En esta fase el analista deberá realizar el Metamodelo de Contexto de Usuario (UCM, por sus siglas en inglés), representando mediante un hexágono a los *stakeholders* que estarán involucrados en el desarrollo del *software* (ver Figura 2.6).
4. Fase 4: Análisis basado en el contexto de los requisitos de aceptación. En esta fase, *Agon-Tool* analizará el UCM creado anteriormente y realizará sugerencias al analista con el fin de mejorarlo (ver Figura 2.7).

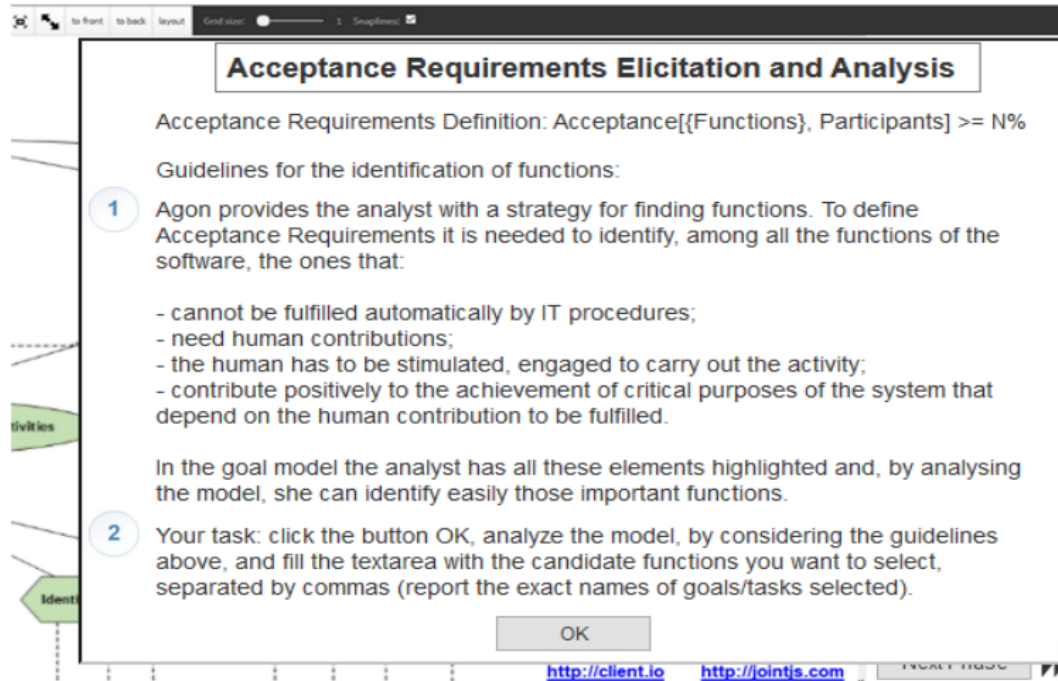


Figura 2.4. Elicitación y análisis de requisitos con Agon-Tool (tomada de Piras (2018))

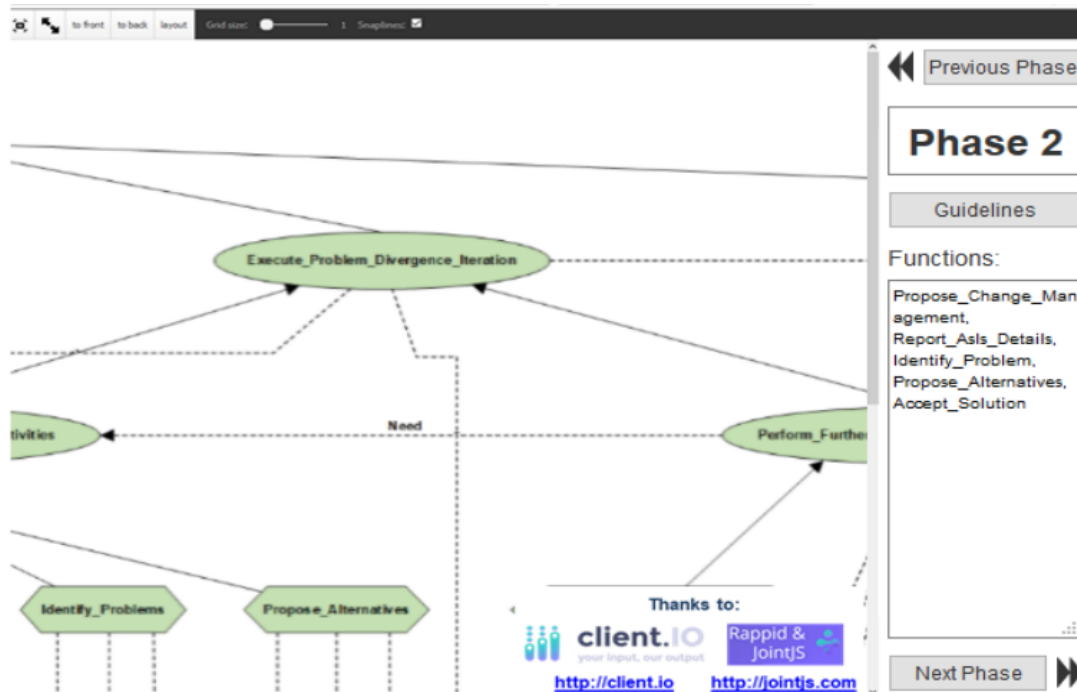


Figura 2.5. Registro de requisitos con Agon-Tool (tomada de Piras (2018))

5. Fase 5: Refinamiento de los requisitos de aceptación. El objetivo principal de esta fase es recomendar tácticas al analista que pueda utilizar para desarrollar el metamodelo (ver Figura 2.8). Así, se consideran las siguientes dos subfases:

- Subfase 5a: *Agon-Tool* le proporcionará tácticas al analista con las cuales podrá mejorar el metamodelo. Una vez que el analista las haya analizado, deberá decidir cuáles aceptar y rechazar.
- Subfase 5b: *Agon-Tool* también le proporcionará al analista tácticas adicionales con el fin de alcanzar un metamodelo adecuado (i.e., requisitos de aceptación que ayuden a mejorar la percepción e interacción de los *stakeholders*). Dichas tácticas de aceptación serán opcionales.

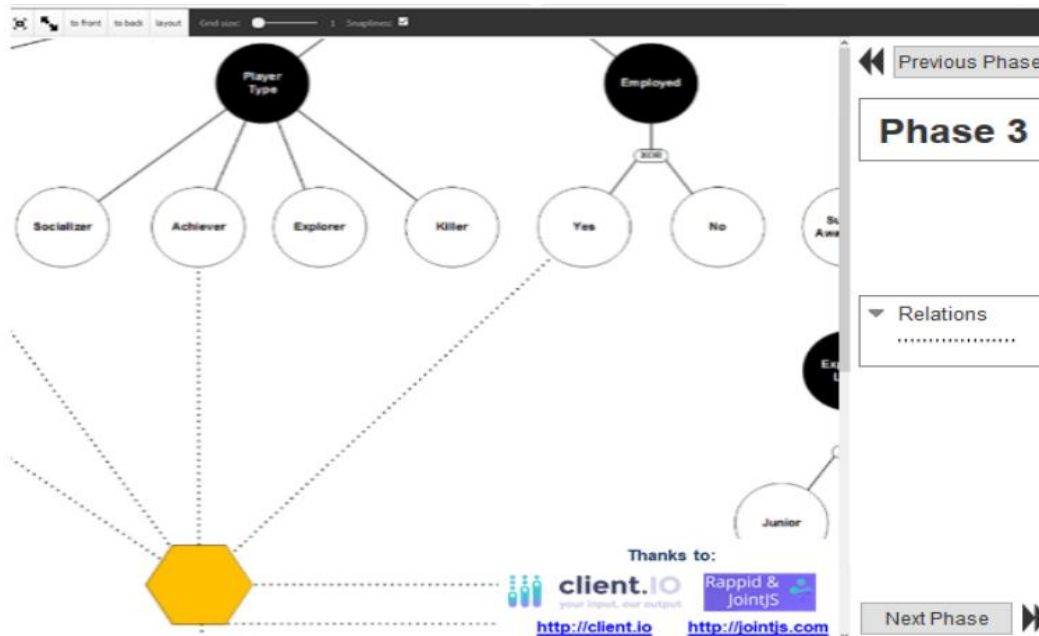


Figura 2.6. Caracterización del contexto con *Agon-Tool* (tomada de Piras (2018))

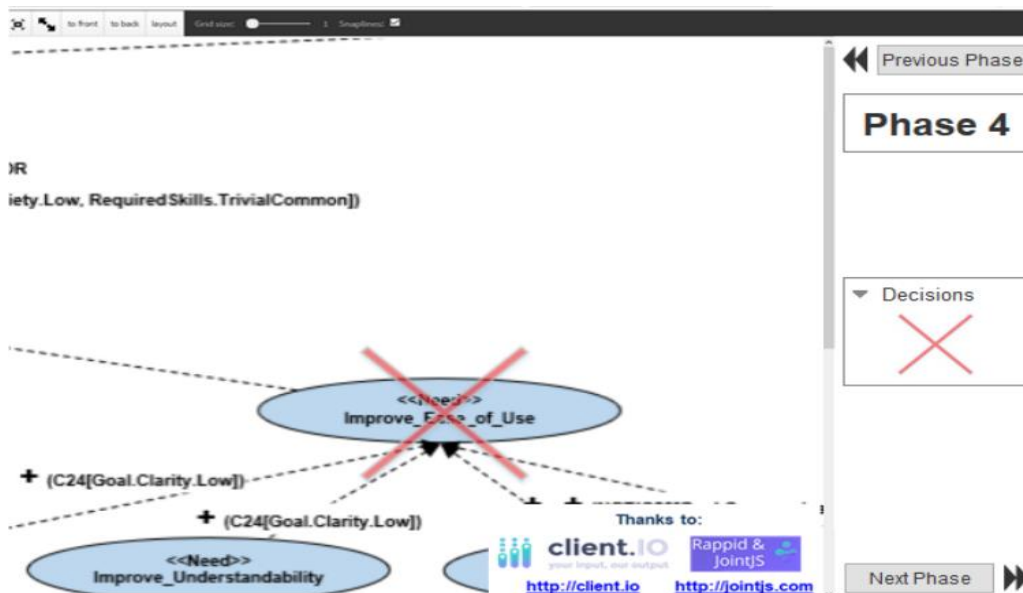


Figura 2.7. Análisis del contexto de los requisitos con *Agon-Tool* (tomada de Piras (2018))

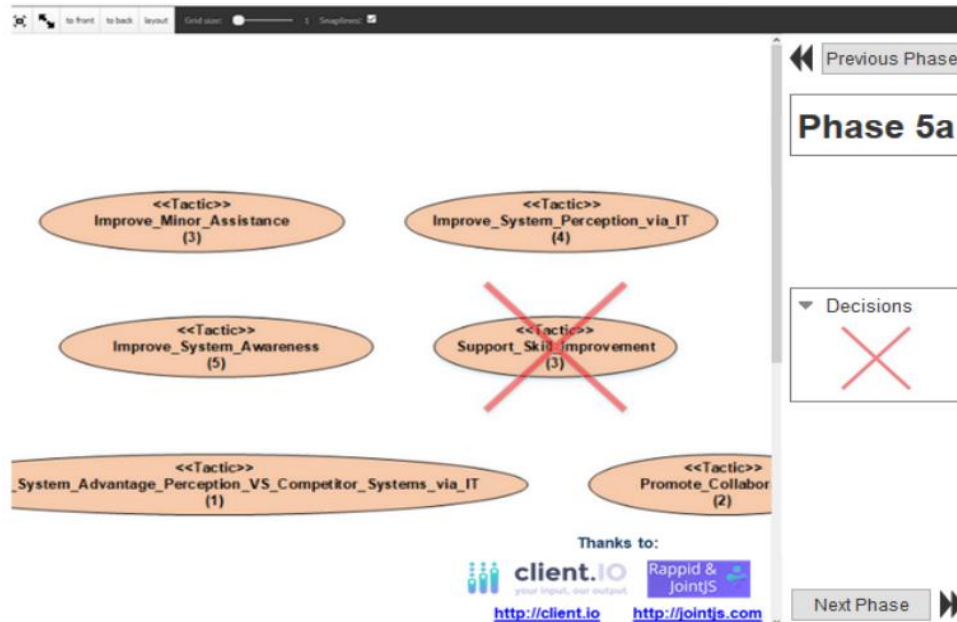


Figura 2.8. Refinamiento de los requisitos con *Agon-Tool* (tomada de Piras (2018))

6. Fase 6: Operacionalización basada en el contexto a través de la gamificación. En esta fase se implementan los aspectos relacionados con la gamificación con el fin de estimular al analista a seguir aprendiendo a través de acciones interactivas, como el uso de un objetivo con movimientos (e.g., la acción de arrastrar el símbolo de tachado - X - desde el recuadro de decisiones). Por lo tanto, *Agon-Tool* proporciona elementos visuales e interactivos al analista para obtener el UCM. En este punto se debe decidir cuáles decisiones se implementarán y, en caso de que no se esté de acuerdo, cuáles eliminar. Una vez que el analista haya aceptado o rechazado las sugerencias deberá avanzar a la siguiente etapa (ver Figura 2.9).

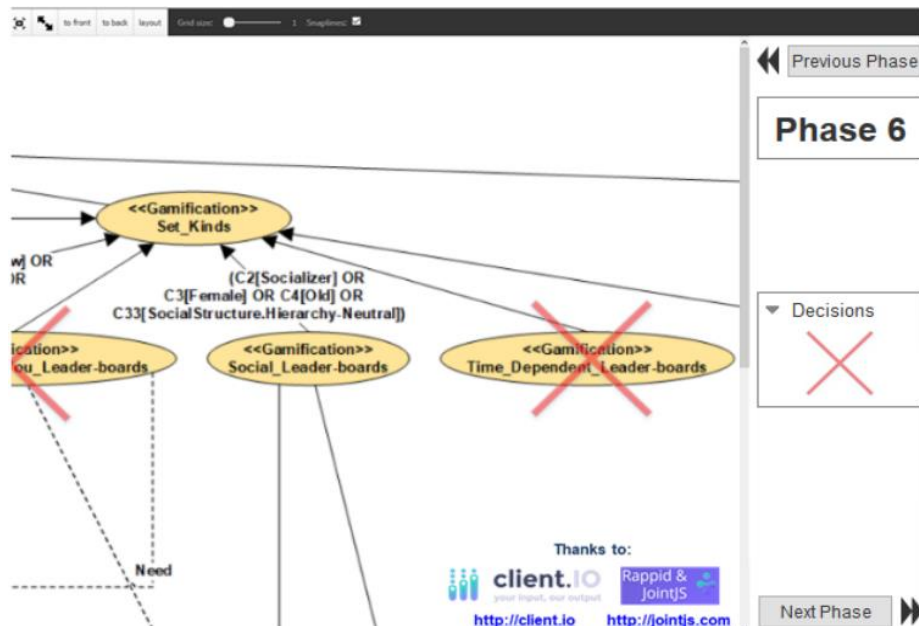


Figura 2.9. Uso de la gamificación para operacionalizar los requisitos con *Agon-Tool* (tomada de Piras (2018))

7. Fase 7: Instanciación dependiente del dominio de los mecanismos de incentivos. Esta fase es concerniente a la instanciación de la gamificación realizada por el analista sobre la solución propuesta por *Agon-Tool*. En este sentido, esta última fase tiene una relevancia importante puesto que los metamodelos generados por *Agon-Tool* son genéricos y no se refieren a un dominio en particular. Por lo tanto, es responsabilidad del analista adaptar la solución de gamificación generada a las limitaciones de su dominio. Una vez realizada esta adaptación, el analista obtendrá el UCM definitivo que corresponda los requisitos elicitados (ver Figura 2.10).

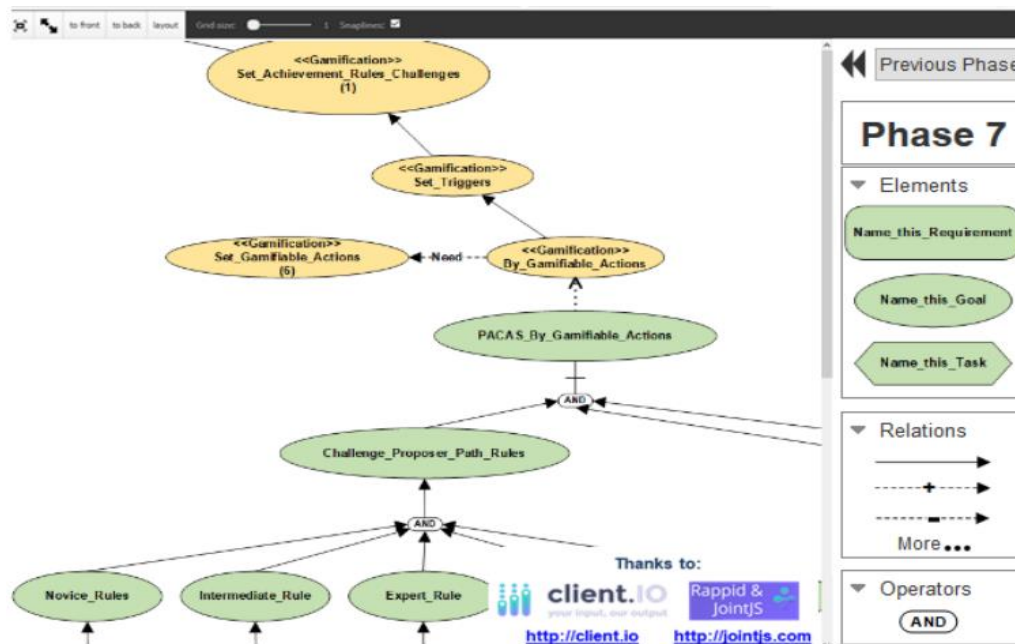


Figura 2.10. Instanciación de la gamificación con *Agon-Tool* (tomada de Piras (2018))

2.4.3.2.3. Resultados

Se realizaron diferentes evaluaciones cualitativas y cuantitativas sobre *Agon-Tool*, las cuales consideraron diferentes escenarios y objetivos, concretamente los siguientes: (1) gamificación de un sistema de planificación de reuniones para motivar la participación de los usuarios en la organización de las mismas, y (2) gamificación de un sistema de gestión de alimentos de una tienda inteligente. Las evaluaciones condujeron a la obtención de los siguientes resultados:

- Las experiencias de 27 participantes (entre expertos y estudiantes de maestría) relacionadas con la aplicación de la gamificación para la elicitación de los requisitos de ambos sistemas de *software*, a través de *Agon-Tool*, fueron comparadas. Lo(a)s participantes fueron entrenados mediante clases presenciales y empleando *Agon-Tool*. Al final de la actividad práctica, todos lo(a)s participantes expresaron su preferencia por *Agon-Tool*.
- Aunado a lo anterior, *Agon-Tool* se empleó para el análisis y diseño de una plataforma de *software* que apoyara las actividades de toma de decisiones en el contexto del proyecto Europeo denominado “Gestión Participativa del Cambio de la Arquitectura de Sistemas ATM” (PACAS, por sus siglas en inglés), el cual se relaciona con la gestión de cambios en los sistemas que utilizan los cajeros automáticos o mejor conocidos como sistemas de modo de transferencia asíncrona (ATM, por sus siglas en inglés). Específicamente, la plataforma creada

sirvió para motivar a los tomadores de decisiones a colaborar de manera participativa mediante el uso de dicho sistema de *software*. Todo(a)s lo(a)s participantes concordaron en que la gamificación les facilitó promover la colaboración durante la elicitación de los respectivos requisitos del *software*.

- Los participantes también afirmaron haber recibido orientación clara durante las sesiones de trabajo. En particular, aquello(a)s participantes que carecían de experiencia en el dominio donde se aplicarían los *software* que se desprenderían de las sesiones de elicitación, afirmaron que apreciaron que el enfoque les proporcionó la orientación precisa al resaltar todos los aspectos más importantes cuando se realizaron las discusiones de análisis y diseño. Refirieron además haber recibido, durante todas las fases, sugerencias relevantes como factores psicológicos y conceptos de gamificación que se ajustan a la caracterización de los usuarios y su contexto. Finalmente, esto(a)s participantes informaron que al utilizar *Agon-Tool* se redujo el esfuerzo total necesario para analizar y diseñar una solución basada en la gamificación, puesto que les ofreció un conjunto más amplio de estrategias psicológicas y mejores prácticas que les permitieron elucidar ideas que un analista pudiera tener en mente.

2.4.4. El uso de los juegos serios en la enseñanza/aprendizaje de la elicitación de requisitos

De manera similar, también existen algunas propuestas relacionadas con el desarrollo de JS que se enfocan en brindar un soporte educativo para la enseñanza/aprendizaje de los procesos de la IR. A continuación se presentan algunos ejemplos de juegos que fueron creados para complementar el aprendizaje en la ER.

2.4.4.1. El aprendizaje activo de la elicitación de requisitos de *software* con LEGO

2.4.4.1.1. Objetivo

De acuerdo con Kurkovsky et al. (2019), la enseñanza de la IR a nivel licenciatura no es una tarea fácil, ya que a veces puede traer consigo inconvenientes derivados, principalmente, de la diversidad de problemas que existen en el área (e.g., aplicación de las técnicas de elicitación, la identificación de los *stakeholders*, la clasificación de los requisitos), temas que no pueden abordarse aplicando un único enfoque debido a los diferentes dominios de conocimiento. Considerando lo anterior, es necesario proporcionar a los estudiantes el conocimiento y la experiencia suficientes para que tengan una idea en general de cómo resolver cualquier tipo de problema que se les presente dentro de la industria. Por tal motivo, se ha buscado implementar JS que les permitan obtener mayor experiencia por medio de escenarios que representan situaciones y problemas cercanos a los que se encuentran en la industria. Por lo tanto, se creó un JS no digital que hace uso de bloques de LEGO para aprender a realizar la elicitación correcta de los requisitos de *software*, considerando un problema específico que es proporcionado por los profesores.

2.4.4.1.2. Descripción

El JS está conformado por dos módulos que se relacionan con las actividades que conforman a la ER y que se describen a continuación:

- Módulo 1: Elicitación de requisitos y refactorización de un sistema en un entorno de desarrollo ágil. Lo(a)s estudiantes deberán trabajar en equipos de 5 integrantes con el fin de implementar un prototipo de un sistema que, en el contexto del juego, se les presenta como una casa. Con este módulo se busca alcanzar los siguientes objetivos de aprendizaje:
 - Diseñar un conjunto de preguntas con el fin de elicitar correctamente los requisitos.

- Interactuar con los *stakeholders* con el fin de elicitar los requisitos.
- Entender la importancia del trabajo en equipo y el desarrollo ágil.
- Validar el conjunto de requisitos obtenidos.

De manera general, para realizar correctamente este módulo, los estudiantes deberán realizarle preguntas al *stakeholder* (i.e., rol que desempeña el profesor) con el fin de elicitar los requisitos que harán posible la construcción de una casa. Esta actividad debe realizarse en 2 a 3 *sprints*, para que posteriormente los estudiantes presenten en clase su avance sobre la implementación de su casa con los bloques de LEGO. El *stakeholder* deberá proporcionar realimentación sobre el grado de satisfacción sobre los requisitos obtenidos. Cada equipo cuenta con un kit de LEGO *Serious Play*, el cual contiene todo lo necesario para construir una casa (e.g., ladrillos, arboles, ventanas) una vez que los requisitos de los *stakeholders* hayan sido elicitados mediante el uso de notas adhesivas, bolígrafos, un pizarrón, plumones, etc. Durante el primer *sprint*, los estudiantes plantean de 3 a 4 preguntas que les permitan elicitar los requisitos. Posteriormente, deben escribirlas en el pizarrón para interactuar con el *stakeholder*, el cual responderá dependiendo la manera en que la pregunta fue formulada (e.g., si el equipo escribió una pregunta dicotómica con respuestas Sí o No, los *stakeholders* responderán únicamente con Sí o No). Por lo tanto, es importante que los estudiantes planteen correctamente las preguntas con el fin de obtener la mayor cantidad de información posible. Cuando los requisitos han sido elicitados, los estudiantes pueden construir su primer prototipo de casa. Una vez terminado el primer *sprint*, cada equipo presenta sus avances y el *stakeholder* deberá revisarlos y expresar, de manera general, si sus necesidades iniciales fueron satisfechas o no.

En el segundo *sprint*, el *stakeholder* proporcionará un nuevo requisito y cada equipo deberá implementarlo, a la par de corregir y cumplir los requisitos que no se elicitaron con éxito en el *sprint* anterior. Al final de este *sprint* cada equipo proporcionará una explicación del trabajo realizado. En caso de que los estudiantes necesiten más tiempo o se quiera agregar un nuevo requisito por parte del *stakeholder*, se puede llevar a cabo otro *sprint*, sin embargo, esto no es obligatorio.

Si el tercer *sprint* es llevado a cabo, los estudiantes pueden reflexionar más sobre cómo evolucionaron sus estrategias de refactorización, así como sobre cualquier problema de cobertura/precisión que haya sido descubierto en los requisitos. La opción de diseño para designar un tercer *sprint* opcional es brindar flexibilidad, ya que algunas clases pueden realizarse durante 50 minutos, mientras que otras pueden hacerlo durante 80 minutos. Por lo tanto, este enfoque sugiere lo siguiente:

- La gestión del tiempo es especialmente necesaria en una clase de 50 minutos, por lo que contar con un cronómetro o reloj ayudará a controlar el tiempo.
 - En el caso de una clase más larga, es necesario llevar un registro de quién hizo qué preguntas o qué respondió el profesor. Por lo tanto, se puede utilizar un bloc de notas, una tableta digital o algún otro mecanismo para registrar la información.
- Módulo 2: En este módulo, los estudiantes trabajan en equipo para crear prototipos que compitan contra un sistema de redes sociales bien conocido (e.g., *Facebook*, *Twitter*, *Instagram*, *SoundCloud*) o un sistema de *hardware* bien definido (e.g., aspiradora *Roomba*, un termostato *Nest* o *Echo* de *Amazon*). Los estudiantes intercambian ideas e identifican un conjunto de características que implementan la funcionalidad central y ayudarán a que un

nuevo sistema se distinga de la competencia. Los objetivos de aprendizaje de este módulo incluyen:

- Sintetizar un conjunto básico de requisitos del sistema.
- Generalizar un conjunto de requisitos mediante la identificación de requisitos relacionados y superpuestos.
- Distinguir entre requisitos funcionales, no funcionales y de dominio.
- Validar los requisitos de integridad y consistencia.

Los requisitos previos de este módulo incluyen una familiaridad básica con los requisitos del *software*, así como una exposición previa a los principios de *LEGO Serious Play*. Esta actividad dura unos 60 minutos y funciona bien para un solo equipo de hasta 25 participantes. Los materiales y suministros necesarios para llevar a cabo esta actividad incluyen un juego de LEGO para cada estudiante, así como una “bolsa de exploración” de *LEGO Serious Play* o *LEGO Classic*, *LEGO City*, *LEGO Friends* o un set similar. Alternativamente, los estudiantes pueden compartir un set de LEGO que incluya diversos ladrillos y minifiguras con accesorios. También se necesitarán algunas notas adhesivas y marcadores. Cada estudiante requiere de un pequeño espacio de construcción separado con una mesa adicional para colocar todos los modelos. El módulo consta de dos partes: elicitación de requisitos (usando *LEGO Serious Play* para la generación de la idea) y análisis de requisitos (sin usar LEGO).

Al realizar las actividades de este módulo con un grupo de estudiantes, es importante tener en cuenta algunas sugerencias prácticas:

- Está bien proponer requisitos similares durante la primera fase (i.e., elicitación). De hecho, es bastante útil tener requisitos similares y/o duplicados para fortalecer el aprendizaje.
- Es importante estar atento al reloj durante la segunda parte del ejercicio (i.e., análisis) para asegurarse de que haya suficiente tiempo para cada parte de esta actividad. La clave para lograr los puntos de aprendizaje está en la realización de discusiones.

2.4.4.1.3. Resultados

La evaluación empírica del JS se realizó en una universidad pública regional ubicada en el noreste de los E.E.U.U., con la participación de 21 estudiantes inscritos en un curso de IS a nivel licenciatura. Dicha evaluación se basó en el uso del Cuestionario de Estrategias Motivadas para el Aprendizaje (MSLQ, por sus siglas en inglés), el cual contiene 44 ítems que deben valorarse considerando una escala tipo *Likert* de 5 puntos que van desde “Totalmente en desacuerdo” hasta “Totalmente de acuerdo”, mediante el cálculo de la media (M) y la desviación estándar (SD). El MSLQ mide la orientación motivacional del estudiante, el compromiso cognitivo y la autorregulación en el aula. Las variables cognitivas y la autorregulación son características de los estudiantes que eligen involucrarse más cognitivamente. Adicionalmente, el MSLQ evalúa cinco subescalas: ansiedad ante los exámenes, autorregulación, autoeficacia, uso de estrategias cognitivas y valor intrínseco. La validación empírica también incluyó relaciones significativas con el rendimiento académico previo y posterior de los estudiantes participantes. Finalmente, se definieron tres preguntas abiertas que evaluaron aspectos relevantes del proceso de construcción de los modelos, el pensamiento abstracto y el uso de modelos LEGO como representaciones de ideas abstractas.

Considerando esta preparación, los estudiantes fueron evaluados tomando en cuenta las puntuaciones de la subescala del MSLQ, la puntuación general del MSLQ y los promedios de las

calificaciones acumuladas (GPA, por sus siglas en inglés). Un estudiante de cada dos fue asignado aleatoriamente a uno de dos grupos: LEGO o control. Dado que participó un total de 21 estudiante, el estudiante restante fue asignado al grupo LEGO. Durante la realización de los módulos, los dos grupos trabajaron en salas separadas. Una sala incluyó al grupo LEGO y al profesor del curso principal; mientras que el grupo de control se reunió en una habitación separada con un segundo profesor. Cada profesor siguió guiones idénticos con la única excepción de que uno trabajó con LEGO y el otro lo hizo con un pizarrón convencional. En la siguiente reunión de clase, el grupo LEGO se reunió con el *stakeholder* quien respondió a una serie de preguntas abiertas en un formato de *focus group*. Al mismo tiempo, el grupo de control tuvo una sesión informativa con su profesor.

Cuatro semanas después se administró un examen parcial en papel a toda la clase. Dos elementos del examen midieron la capacidad de los estudiantes para identificar los requisitos del sistema, resolver ambigüedades y expresarlos en forma de historias de usuario. El profesor del curso calificó ciegamente los elementos de mitad de período utilizando una escala de 0 a 10, siendo 10 una puntuación perfecta. Al final del semestre, ambos grupos volvieron a responder el MSLQ.

Los resultados obtenidos mostraron que ambos grupos de estudiantes compartieron la misma tendencia entre las subescalas del MSLQ para la evaluación inicial. El valor intrínseco global fue el constructo más alto ($M= 4.31$, $DS= 0.38$), seguido de la autoeficacia ($M= 3.66$, $DS= 0.65$). El uso de estrategias cognitivas ($M= 3.54$, $DS= 0.35$) y la autorregulación ($M= 3.36$, $DS= 0.34$) se calificaron más bajo; sin embargo, las medias de ambas subescalas cayeron en la categoría “De acuerdo” de la escala *Likert*. La subescala más baja fue la ansiedad ante los exámenes ($M= 2.75$, $DS= 1.04$), puesto que ambos grupos de estudiantes tendieron a no calificarse como ansiosos.

Estas similitudes se mantuvieron después de la intervención, arrojando diferencias no significativas tanto en la pre como en la post evaluación, así como entre los dos grupos usando intervalos de confianza *bootstrap* y un nivel de significancia de 0.05. Específicamente, el valor intrínseco para el grupo LEGO ($M= 4.22$, $DS= 0.42$) y el grupo de control ($M= 3.84$, $DS= 0.43$) fueron nuevamente los valores más altos. Las diferencias no significativas para los miembros del grupo LEGO en autoeficacia ($M= 3.80$, $DS= 0.61$), uso de estrategias cognitivas ($M= 3.60$, $DS= 0.47$) y autorregulación ($M= 3.42$, $DS= 0.42$) fueron similares a las medias del grupo de control ($M= 3.47$, $DS= 0.60$), ($M= 3.56$, $DS= 0.36$) y ($M= 3.31$, $DS= 0.68$), respectivamente. Si bien la ansiedad general ante los exámenes aumentó ligeramente después ($M= 3.02$, $SD= 0.99$), las puntuaciones medias del grupo LEGO ($M= 2.82$, $DS= 1.11$) fueron similares a las puntuaciones del grupo de control ($M= 3.23$, $DS= 0.85$).

Como medida del rendimiento académico, los dos ítems parciales reflejaron el contenido del módulo. Ambos grupos de estudiantes exhibieron desempeños fuertes similares no estadísticamente significativos con medias superiores a 9.0 para ambos grupos en ambos ítems, ningún estudiante recibió una puntuación inferior a 7.0 para cualquiera de los ítems.

En resumen, la evaluación del impacto en el aprendizaje de lo(a)s estudiantes mostró su entusiasmo en el uso de módulos de aprendizaje basados en LEGO dentro del aula, mientras que sus comentarios resaltaron una serie de beneficios tangibles que los ayudaron a alcanzar sus objetivos de aprendizaje.

2.4.4.2. Diseño y desarrollo de un juego serio para la enseñanza de la elicitación y el análisis de requisitos

2.4.4.2.1. Objetivo

La elicitación y el análisis de los requisitos son dos pasos importantes para el desarrollo exitoso de un *software*, ya que cualquier error cometido en estos procesos puede generar consecuencias negativas, dando como resultado la implementación de un proyecto fracasado. Por tal motivo, es

necesario asegurar que la enseñanza-aprendizaje de la IR se lleve a cabo de manera correcta, con el fin de mejorar la comprensión, interés y participación de los estudiantes. En este sentido, Ibrahim et al., (2019) propusieron un JS basado en el modelo ADDIE (Análisis, Diseño, Desarrollo, Implementación y Evaluación), que tiene como objetivo aumentar el interés de lo(a)s estudiantes en la IR y ayudarlo(a)s a mejorar su comprensión de la elicitación y el análisis de los requisitos.

2.4.4.2.2. Descripción

El diseño del JS consideró las recomendaciones de Frederick Sheldon⁹ sobre las actividades que deben realizarse para aprender a elicitar y analizar un conjunto de requisitos de *software*, las cuales se resumen mediante la siguiente definición: “*Describir las actividades en el proceso de Ingeniería de Requisitos que se ocupan de descubrir y analizar requisitos considerando su completitud, consistencia, relevancia y practicidad, así como negociar los requisitos finales para el desarrollo del sistema*”. Por consiguiente, se consideraron los siguientes 6 objetivos de aprendizaje para abordar dichas actividades con el JS:

- Descubrir los requisitos.
- Analizar la completitud de los requisitos.
- Analizar la relevancia de los requisitos.
- Analizar la consistencia de los requisitos.
- Analizar la practicidad de los requisitos.
- Negociar los requisitos finales del sistema.

Antes de usar el juego, lo(a)s estudiantes reciben los conocimientos básicos sobre la elicitación y el análisis de requisitos a través de un enfoque convencional (i.e., clases presenciales). Posteriormente se continua con las sesiones de juego para poner en práctica los conocimientos adquiridos en los diferentes escenarios del juego. Finalmente, se genera una discusión colaborativa entre lo(a)s estudiantes sobre los resultados obtenidos en el juego. En este sentido, el JS está compuesto por tres niveles que exploran los objetivos de aprendizaje mencionados anteriormente:

- El primer nivel se desarrolla en un escenario donde el(la) jugador(a) debe elicitar los requisitos para el desarrollo de una aplicación que funja como un asistente de investigación.
- El segundo nivel está representado por el escenario de un centro de servicio automotriz donde el(la) jugador(a) debe elicitar los requisitos para un sistema de reserva de citas.
- El tercer nivel es un escenario de supermercado donde el(la) jugador(a) necesita elicitar los requisitos para un sistema de inventario.

Es importante aclarar que mientras más alto sea el nivel en el que se encuentre el(la) estudiante, más serán los desafíos que afrontará para lograr sus objetivos. Como primer paso dentro del juego, el(la) estudiante deberá asumir el rol de analista de requisitos y deberá elicitar los requisitos para el desarrollo del *software* según los niveles antes mencionados. Los(las) estudiantes deberán recorrer los diferentes escenarios del JS en busca de los *stakeholders* o personajes no jugables (NPC, por sus siglas

⁹ Frederick T. Sheldon es profesor en la Universidad de Idaho, E.E.U.U, con más de 35 años de experiencia profesional en la academia, la industria y el gobierno trabajando en la resolución de diversos problemas dentro del alcance de la Ingeniería de *Software*, los métodos formales y el aseguramiento de la calidad.

en inglés), los cuales están divididos en dos grupos: los que proporcionan requisitos y los que proporcionan misiones. Dependiendo del escenario en el que se encuentre el(la) estudiante, los *stakeholders* irán cambiando (e.g., ejecutivos, líderes de proyecto, cajeros).

Para cumplir con los objetivos de aprendizaje, los requisitos proporcionados por todos los NPC se clasifican en función de su integridad, consistencia, relevancia y practicidad, y el(la) estudiante deberá etiquetar las categorías correctamente. En el nivel 1, se le pregunta al estudiante si los requisitos están completos y son relevantes. La consistencia se agrega a partir del nivel 2, mientras que en el nivel 3 se requiere que la practicidad de los requisitos sea evaluada.

El juego comienza después de que el estudiante escoge la opción *Play* del menú principal. Posteriormente, se instruye al estudiante para que vaya en busca del *stakeholder* que le proporcionará su primera misión. Una vez aceptada la misión, el tiempo empieza a correr y el(la) estudiante dispondrá de un total de 600 segundos para completarla (ver Figura 2.11). Dado que son varios los NPC que están en cada escena, la información relevante de éstos, así como su rol y descripción de trabajo, ser visualiza en un cuadro de diálogo cada vez que el estudiante se encuentre con ellos (ver Figura 2.12). Esto le permite al estudiante decidir con qué NPC hablar y se fomenta, al mismo tiempo, que éste(a) practique para determinar qué personas son adecuadas para que proporcionen los requisitos. De hecho, PNC ficticios fueron incluidos deliberadamente en el JS por este motivo. Dado que los NPC proporcionan cada requisito en forma de cuadro de diálogo, el(la) estudiante tendrá la opción de aceptar el requisito proporcionado por el NPC, rechazarlo por ser irrelevante, o marcarlo como incoherente o poco práctico, de acuerdo con el nivel de juego. El diseño del JS incluyó un botón en la parte superior izquierda de la pantalla para mostrarle a lo(a)s estudiantes todos los requisitos que se hayan recopilado de los NPC.

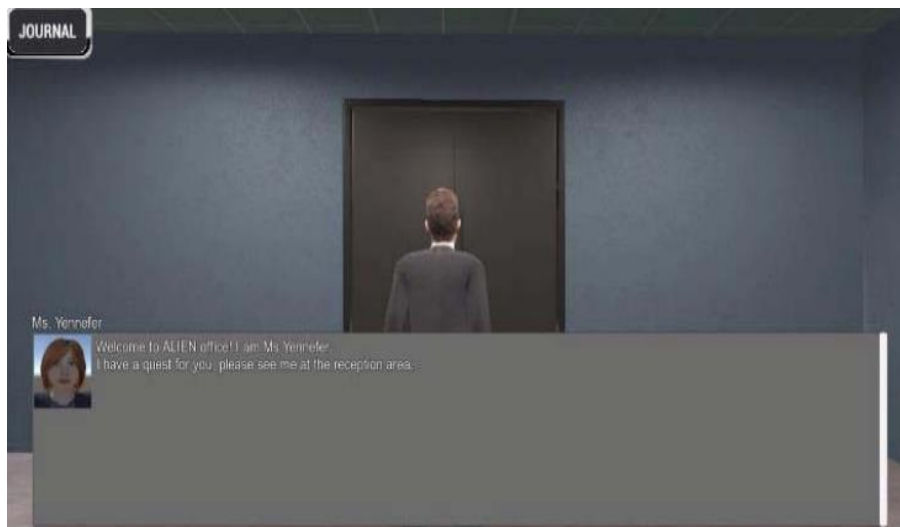


Figura 2.11. Instrucciones recibidas para la realización de la misión (tomada de Ibrahim et al., (2019))

Una vez terminada la misión (i.e., proceso de elicitación de requisitos), el(la) estudiante tendrá que volver con el *stakeholder* que se la asignó, para que éste le proporcione un puntaje. En caso de que el(la) estudiante reciba un puntaje alto pasará al próximo nivel, pero en caso contrario, el(la) estudiante perderá y el juego terminará, por lo que tendrá que reiniciar el nivel en busca de la puntuación mínima aprobatoria para pasar al siguiente nivel. Otro factor importante por recordar es el tiempo, ya que, si no se lleva a cabo la elicitación de requisitos en el tiempo estimado, el juego tendrá que reiniciarse.



Figura 2.12. Información del *stakeholder* y el trabajo que desempeña (tomada de Ibrahim et al., (2019))

Las puntuaciones dentro del JS se otorgan de la siguiente manera:

- Se otorgarán puntos si el(la) estudiante identifica correctamente a los *stakeholders*.
- Se otorgará un premio si el(la) estudiante ignora correctamente a los *stakeholders* que no debe tomar en cuenta.
- La calificación final se asignará dependiendo de la cantidad de requisitos elicitados correctamente. Por lo tanto, no se asignará calificación si el(la) estudiante incluye requisitos ficticios.

2.4.4.2.3. Resultados

La evaluación empírica del JS se llevó a cabo en la Universidad Nacional de Tenaga, Malasia, considerando únicamente el primer nivel. Esta decisión llevó a que solamente los primeros tres objetivos de aprendizaje fueran evaluados. Dicha evaluación consideró a un total de 10 estudiantes de licenciatura y pretendió obtener una realimentación acerca de sus logros con respecto a la enseñanza proporcionada por el juego. Tres de los estudiantes ya habían completado un curso sobre “Análisis y Diseño de Sistemas” (SAD, por sus siglas en inglés), otros tres habían completado solamente el curso de “Fundamentos de Ingeniería de *Software*” (FSE, por sus siglas en inglés) y los cuatro restantes habían completado los cursos de FSE e Ingeniería de Requisitos. Todo(a)s los participantes tardaron entre 144 y 454 segundos en completar el juego, siendo el tiempo promedio de 314.6 segundos.

El desempeño de cada participante fue evaluado con respecto al número total de requisitos elicitados, el número de requisitos elicitados que fueran correctos, y el número de NPC seleccionados. Los requisitos correctos se dividieron en correctos positivos y correctos negativos. Correcto positivo significa que los requisitos del sistema a desarrollar en el juego fueron correctamente seleccionados por el(la) estudiante. Negativo correcto significa que los requisitos ficticios en el juego fueron descartados correctamente por el jugador. También se aplicaron definiciones similares a los NPC. Es decir, los NPC correctos positivos son los NPC correctos que el(la) estudiante identificó y de los que obtuvo los requisitos, mientras que los NPC correctos negativos son los ficticios que los jugadores decidieron ignorar al jugar.

Los resultados obtenidos demostraron que la mayoría de lo(a)s estudiantes seleccionaron los cinco NPC correctos en el juego. Sin embargo, una revisión más profunda evidenció que en realidad estaban recopilando requisitos de todos los NPC en el juego, incluidos los ficticios. De hecho, solamente dos participantes lograron ignorar correctamente a los NPC ficticios. Por otro lado, con relación al primer objetivo de aprendizaje (*Descubrir los requisitos*), los resultados mostraron que los estudiantes tuvieron problemas para identificar a los NPC correctos de los que debían elicitarse los requisitos. Por lo tanto, el primer objetivo de aprendizaje no se logró satisfactoriamente. No obstante, un análisis posterior demostró que podría haber un problema de diseño en el JS, tal que no fuera posible distinguir claramente a los NPC relevantes e irrelevantes. Con respecto al número de requisitos correctos que fueron elicitados por lo(a)s estudiantes, medida para evaluar el segundo objetivo de aprendizaje (*Analizar la completitud de los requisitos*), se observó que la mayoría de los participantes lograron identificar correctamente al menos cinco de los nueve requisitos correctos. En este sentido, el porcentaje de requisitos correctos recogidos fue del 74.4%. Finalmente, los resultados obtenidos para evaluar la consecución del tercer objetivo de aprendizaje implicó la capacidad de lo(a)s estudiantes de *analizar la relevancia de los requisitos*. A diferencia del segundo objetivo de aprendizaje, el tercer objetivo midió la capacidad de lo(a)s estudiantes de identificar requisitos irrelevantes. Así, de un total de 12 requisitos correctos en el juego, que incluían nueve requisitos positivos correctos y tres negativos correctos, nueve de los 10 estudiantes participantes seleccionaron al menos cinco de los nueve requisitos relevantes. De hecho, cuatro de éstos(a)s pudieron seleccionar correctamente ocho requisitos relevantes. Estos resultados indicaron que la mayoría de lo(a)s estudiantes pudieron determinar los requisitos relevantes en el escenario del juego. Sin embargo, tres participantes no rechazaron ninguno de los requisitos irrelevantes, cuatro rechazaron solo uno y dos participantes rechazaron dos. Solo un estudiante pudo descartar los tres requisitos irrelevantes. Por lo tanto, se observó un mejor desempeño con respecto a la identificación de los requisitos correctos en comparación con la identificación de requisitos incorrectos o irrelevantes.

Al finalizar la evaluación empírica se concluyó que los escenarios interactivos proporcionados por el JS permitieron a lo(a)s estudiantes tener más oportunidades de practicar sus conocimientos teóricos sobre la elicitación y el análisis de requisitos fuera del horario de clase. De esta manera y con el uso prolongado, lo(a)s estudiantes podrán comprender mejor la elicitación y el análisis de requisitos y, al egresar, podrán realizar efectivamente ambos procesos en un entorno real de trabajo.

2.4.4.3. Desarrollo de un videojuego educativo para la enseñanza/aprendizaje de la clasificación de requisitos en la Ingeniería de Software

2.4.4.3.1. Objetivo

Chanchí et al., (2019) afirmaron que la identificación y clasificación de requisitos tampoco es un tema fácil de tratar dentro de la enseñanza de la IR, por tal motivo propusieron el desarrollo de *GDevelop*, una plataforma de código abierto que está destinada a la creación de videojuegos en dos dimensiones y que se enfoca en usuarios no programadores. Es decir, dicha plataforma permite la creación de videojuegos mediante la estrategia de programación orientada a eventos y acciones, en la cual no se codifican líneas de código, sino que se define el comportamiento de los objetos agregados al escenario mediante eventos y acciones. En este sentido, *GDevelop* fue utilizada para crear un JS cuyo objetivo fue facilitar la identificación y clasificación de requisitos funcionales y no funcionales.

2.4.4.3.2. Descripción

El desarrollo del JS contempló cuatro fases adaptadas de la metodología denominada *Design Thinking* (Henriksen et al., 2017): empatizar y definir, idear, prototipar, y evaluar. El diseño de este juego se basó en la idea central de una interfaz sencilla en la cual los requisitos, ya fueran funcionales

o no funcionales, descendieran de la parte superior de la pantalla y en la parte inferior fueran depositados en alguna de las dos carpetas disponibles. Por cada requisito clasificado de manera correcta, el(la) estudiante irá sumando un punto, obteniendo así al final del juego una relación entre los requisitos acertados y los intentos realizados. La interfaz de alto nivel que representó esta idea se muestra en la Figura 2.13.

Considerando lo anterior, el JS implementó una lógica sencilla para inducir al jugador a distinguir entre requisitos funcionales y no funcionales. Una vez que éste presiona el botón de inicio, se genera un requisito aleatorio a partir de los requisitos definidos y almacenados en el juego, de tal manera que tales requisitos empiezan a descender desde la parte superior de la pantalla. Mientras esto sucede, el(la) jugador(a) debe clasificar a los requisitos como funcionales o no funcionales llevándolos a la carpeta correcta, de tal modo que, en caso de que un requisito haya sido bien clasificado, el JS incrementará el puntaje y el número de intentos, en caso contrario solo incrementa el número de intentos. Cada vez que el puntaje y los intentos son desplegados en la pantalla, el JS verifica si se han cumplido los diez intentos o no, en caso de que no se hayan cumplido se repite el proceso de obtener los requisitos aleatorios, de lo contrario se muestra en pantalla el resultado final y el usuario puede terminar el juego o volverlo a iniciar.

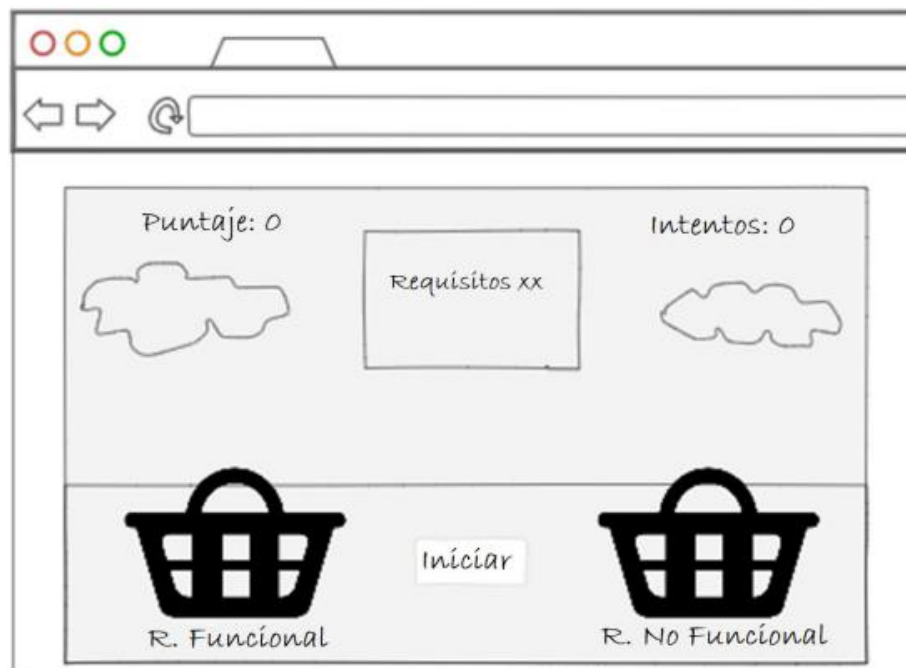


Figura 2.13. Interfaz propuesta para el juego creado con *GDevelop* (tomada de Chanchí et al., (2019))

La Figura 2.14 muestra que los diferentes ítems u objetos del videojuego pueden ser en este caso de dos tipos específicos: *sprites* (i.e., requisito aleatorio, carpetas, botón de iniciar o terminar, fondo del escenario, y césped) o mensajes en pantalla (i.e., puntaje, intentos, mensaje en el fondo de la pantalla). Es importante resaltar que los *sprites* del juego se asocian con imágenes, como es el caso del *post-it* donde se ubican los requisitos o las carpetas en donde el(la) jugador(a) deposita los requisitos funcionales o no funcionales.



Figura 2.14. Interfaz definitiva del juego para clasificar requisitos funcionales y no funcionales (tomada de Chanchí et al., (2019))

Por lo tanto, el JS creado escoge aleatoriamente un conjunto de requisitos ya sean funcionales o no funcionales, previamente definidos. A partir del requisito escogido por el juego de manera aleatoria, el(la) jugador(a) debe depositarlo en la carpeta adecuada antes de que toque el suelo. Siguiendo esta lógica, cada jugador(a) deberá conseguir un puntaje de 10 para ganar, ya que en caso contrario tendrá que repetir el juego.

2.4.4.3.3. Resultados

El JS carece de evaluación empírica con estudiantes reales, puesto que únicamente se realizó una valoración de su usabilidad. En este sentido, la población que valoró la usabilidad del videojuego estuvo conformada por cinco docentes de IS con entre cinco y ocho años de experiencia en la enseñanza de esta área a nivel universitario y con un nivel de conocimiento medio en el área de Interacción Humano Computadora, específicamente en aspectos de usabilidad. Estos profesores realizaron la valoración a través de la aplicación de un método de inspección guiado por las heurísticas de videojuegos (Pinelle et al., 2008). A pesar de no existir una evaluación basada en datos cuantitativos que reflejen el desempeño de los jugadores, la aplicación subjetiva de las 10 heurísticas de Nielsen permitió la obtención de aspectos positivos y mejoras que debían considerarse en una futura versión del JS. En términos generales, estos aspectos buscaron contribuir a mejorar la usabilidad del JS y por ende el nivel de aceptación por parte de lo(a)s estudiantes que lo usarían, de tal modo que éste sea más cercano al usuario y pueda contribuir a mejorar sus habilidades en la clasificación de los requisitos funcionales y no funcionales.

Por consiguiente, la valoración realizada al JS permitió evidenciar, de acuerdo con la opinión de los profesores, que el juego era sencillo e intuitivo para el usuario final. Sin embargo, se destacó también que había ciertos aspectos que debían mejorarse para incorporarlo a un escenario real, como la inclusión de las definiciones de requisito funcional y no funcional en la ayuda del juego, así como la discriminación en el puntaje de los aciertos en los dos tipos de requisitos.

Aunado lo anterior, se recogió evidencia que demostró que *GDevelop* es una plataforma adecuada para el diseño y construcción de recursos educativos que pueden ser utilizados en el aula de clase. Adicionalmente, la plataforma permite la exportación de los videojuegos creados a diferentes plataformas, aportando ventajas de portabilidad y flexibilidad.

2.4.5. El uso de la gamificación y juegos serios en la enseñanza/aprendizaje de la identificación de los *stakeholders* dentro de la Ingeniería de Requisitos

Como se ha mencionado anteriormente, los *stakeholders* son parte crucial de la IR y son, al mismo tiempo, un factor clave para garantizar el éxito en la identificación de las fuentes que proveerán al ingeniero(a) de requisitos de necesidades y limitaciones que le permitan descubrir qué debe hacer exactamente el *software*. Por lo tanto, una correcta identificación conducirá a un involucramiento correcto de los *stakeholders*, aumentando así la probabilidad de que la elicitación de requisitos sea exitosa y, por consiguiente, que se desarrolle un *software* exitoso. En este sentido, de acuerdo con Kolpondinos y Glinz (2020), el uso correcto tanto de los principios de la gamificación como de los JS, proporciona beneficios importantes en la educación de la ER puesto que ambos permiten involucrar a lo(a)s estudiantes más activamente en el aprendizaje de una actividad que requiere altos niveles de compromiso, iniciativa e interacción.

A continuación se presentan dos propuestas que demuestran los beneficios de involucrar a lo(a)s estudiantes en un aprendizaje activo para identificar a los *stakeholders* correctamente.

2.4.5.1. GARUSO: Una plataforma de gamificación para involucrar a los *stakeholders* en un contexto organizacional

2.4.5.1.1. Objetivo

Kolpondinos y Glinz (2020) argumentaron que un factor clave para obtener resultados efectivos de la IR era lograr el involucramiento correcto de los *stakeholders*. Sin embargo, su identificación puede traer consigo varios inconvenientes dado que muchas de las técnicas usadas para realizar ambas tareas (i.e., identificación e involucramiento) asumen que la identificación de los *stakeholders* se restringe únicamente al interior de las organizaciones involucradas en la solicitud, desarrollo y/o mantenimiento del *software*. De hecho, se tiene la creencia de que dichos *stakeholders* siempre contribuirán con necesidades o limitaciones que faciliten el entendimiento de los requisitos del *software*. Lamentablemente, estas suposiciones no necesariamente son válidas para los casos donde la mayoría de los *stakeholders* (i.e., en particular los usuarios y gente afectada por el *software*) están ubicados fuera de la organización (e.g., *Spotify*, *Netflix*, *Amazon Prime*), puesto que no se les conoce, no se les puede identificar fácilmente, y no se les puede obligar a que participen en los procesos de la IR. Considerando este escenario es que se creó GARUSO, una plataforma que tiene como objetivo el ayudar a las personas con la identificación de los *stakeholders* que se encuentran fuera del alcance organizacional.

2.4.5.1.2. Descripción

GARUSO fue creada como una plataforma social que permite que los *stakeholders* que estén fuera del alcance de las organizaciones participen activamente en las actividades de elicitación y priorización de requisitos. Dicha plataforma usa la gamificación para motivar a los *stakeholders* a participar voluntariamente en estas actividades (ver Figura 2.15).

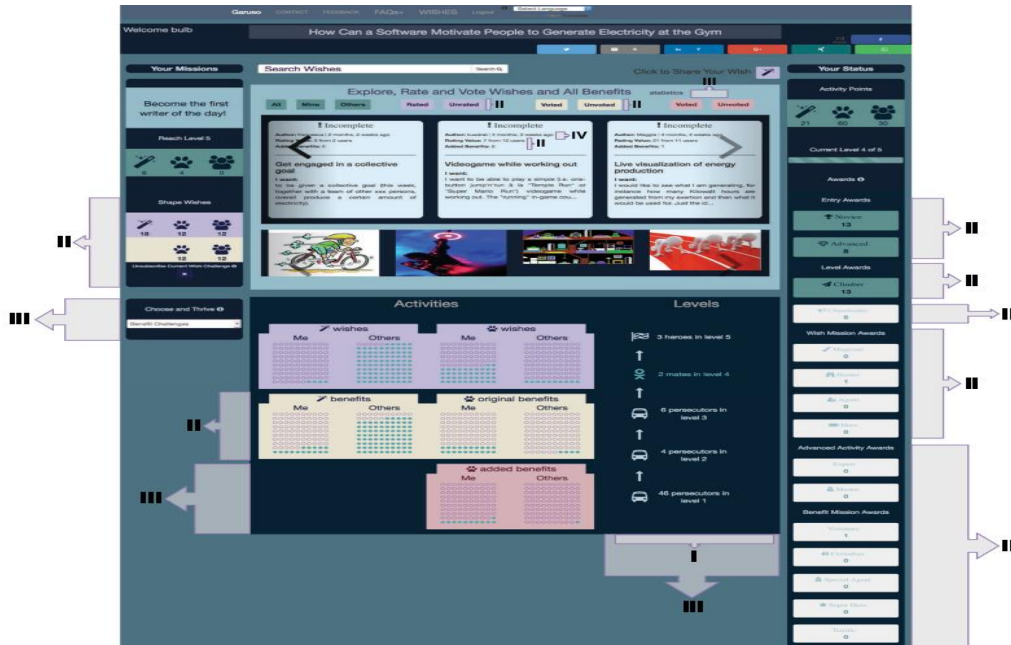


Figura 2.15. Pantalla inicial de GARUSO (tomada de Kolpondinos y Glinz (2020))

La plataforma está conformada por dos componentes principales que se describen a continuación:

- **Descriptor de requisitos.** Este componente está caracterizado por el alcance (*post* o *sub-post*) y la categoría (compartida o evaluada). Mediante un *post*, los usuarios de la plataforma describen un deseo que tengan respecto al *software* que se pretenda desarrollar, así como el beneficio que obtendrán siempre y cuando se llegue a satisfacer dicho requisito. Para completar el *post*, los usuarios deben proporcionar el contexto en el que obtendrán mayor beneficio del requisito a través de una “etiqueta de beneficio” (ver Figura 2.16). Para fortalecer la colaboración, los *stakeholders* podrán agregar *sub-posts* dentro de un *post*, los cuales permiten describir un beneficio adicional del deseo propuesto en el *post*. Además, al compartir esta información, todos los usuarios podrán evaluar *posts* y *sub-posts* de otros *stakeholders*, lo que ayuda y fortalece el proceso de priorización de requisitos. Los *posts* y *sub-posts* permiten construir una historia de usuario que incluya información contextual de la forma “*quiero [deseo] para obtener [beneficio] que más contribuya a [etiqueta]*”, donde esta última parte puede repetirse las veces que sean necesarias.
- **Identificador de *stakeholders*.** GARUSO permite la identificación de *stakeholders* mediante la funcionalidad que le permite a un usuario el compartir información mediante los canales de comunicación de otras plataformas sociales. De esta forma, los *stakeholders* registrados en GARUSO pueden invitar a otros *stakeholders* para que participen en las actividades de elicitación de requisitos.

Por otro lado, GARUSO cuenta con un motor de gamificación que lo vuelve más atractivo para los *stakeholders* y que consta de dos conjuntos de reglas: el sistema de puntos y el sistema de recompensas. Ambos se describen de la siguiente manera:

- **Sistema de puntos.** Este se divide en tres categorías:

- Puntos compartidos. Se ganan cuando los *stakeholders* comparten un *post* o un *sub-post*.
- Puntos de evaluación. Se ganan cuando los *stakeholders* evalúan o votan por primera vez por un *post* o *sub-post* compartido por un *stakeholder* dentro de la comunidad de GARUSO.
- Puntos de comunidad. Se ganan cuando otros *stakeholders* evalúan los *posts* o *sub-posts* de un *stakeholder* en particular, o cuando un *stakeholder* agrega *sub-posts* a un *post* de otro *stakeholder*.

Share Your Wish Below

Write a title that describes your WISH: *

Tell us about your WISH by completing this sentences: 'I want...' *

Specify the BENEFIT you get of your wish by completing this sentences: 'So that...' *

Select the category to which you think your BENEFIT contributes most:

Information Representation Comparison Select/Define others

If you have one, upload an image that clarifies your 'Wish with Benefit'

Browse... No file selected.

Click to Add Your Wish

Figura 2.16. Formato utilizado en GARUSO para compartir un *post* (tomada de Kolpondinos y Glinz (2020))

- Sistema de recompensas. Para medir y aumentar la experiencia de los usuarios se utilizan cinco niveles de experiencia, que van de novato (Nivel I) a experto (Nivel V), y que se basan en los tipos de jugadores descritos por Bartle (1996): *triunfador* (aquel jugador que prefiere actuar en el mundo o sistema), *asesino* (aquel jugador que actúa sobre otros), *socializador* (aquel jugador que interactúa con otros), y *explorador* (aquel jugador que interactúa con el mundo o sistema). Considerado estos niveles, la plataforma proporciona recompensas que se basan en

el concepto de motivación y que se definen por la experiencia y habilidades que demuestre el jugador durante las sesiones de trabajo, tal y como se describe a continuación:

- Nivel I. El *stakeholder* demuestra un nivel alto de autonomía pero está bajo en competencia.
- Nivel II. El *stakeholder* demuestra un nivel medio de autonomía y competencia, pero bajo en su capacidad de establecer relaciones con otros *stakeholders*.
- Nivel III. El *stakeholder* demuestra un bajo nivel de autonomía, alto en competencia, y una capacidad mínima de establecer relaciones con otros *stakeholders*.
- Nivel IV. El *stakeholder* demuestra una capacidad importante para establecer relaciones con otros *stakeholders*.
- Nivel IV. El *stakeholder* no demuestra autonomía pero exhibe una capacidad máxima para establecer relaciones con otros *stakeholders*.













La Figura 2.17 muestra el esquema de visualización que GARUSO utiliza para presentar las recompensas descritas anteriormente. Dicho esquema es utilizado por los usuarios como guía para propiciar la interacción.

2.4.5.1.3. Resultados

Se realizó una evaluación empírica sobre la plataforma de GARUSO desde el 2 de septiembre de 2017 hasta el 2 de diciembre de 2017. El objetivo de esta evaluación fue comprobar si la plataforma era capaz de atraer a una multitud de *stakeholders* fuera del alcance de la organización y motivarlos a participar voluntariamente en las actividades de obtención y priorización de requisitos.

La plataforma estuvo disponible en línea durante estos 92 días para que los usuarios pudieran registrarse y participar consecutivamente en la elicitación y priorización de requisitos para el sistema *SmaWoMo*. Dicho sistema fue parte de un proyecto de vida inteligente basado en la eficiencia energética desarrollado por el Instituto Federal de Investigación en Ciencia y Tecnología de Materiales en Suiza. Uno de los objetivos de este proyecto fue el de transformar en electricidad la energía mecánica que generan las personas al usar equipos de entrenamiento en un gimnasio, por lo que fue necesario desarrollar un sistema de *software* que motivara a las personas durante su entrenamiento y, además, midiera y analizará la información recogida durante dicho entrenamiento. Con el objetivo de buscar participantes, se distribuyeron anuncios en *Facebook*, *LinkedIn* y *Xing* entre las dos primeras semanas, repitiendo la acción durante la semana seis. En la segunda semana, se envió un correo electrónico a 26,000 estudiantes y personal de la Universidad de Zúrich para informar la necesidad de obtener requisitos para el desarrollo de *SmaWoMo* a través de la plataforma GARUSO.

Durante el período de tiempo que se destinó para el registro de usuarios de cualquier parte del mundo, 726 aceptaron la invitación. De esta cantidad inicial, únicamente 244 fueron considerados como *stakeholders* potenciales, de los cuales únicamente 63 terminaron siendo *stakeholders* definitivos. Al final, dicha cifra se redujo a 32 *stakeholders* activos, los cuales contribuyeron en la realización de las actividades establecidas por GARUSO.

Category	Activity	Scope				
		Post (user story)			Sub-Post (additional benefit)	
		Wish Part	Benefit Part		Benefit Part	
		Text	Label	Text	Label	
 Share	Share Post					
	Share Sub-Post					
 Evaluate	Rate Post					
	Vote Post					
	Vote Sub-Post					
 Community	Get Shares					
	Get Evaluations					

Significado de íconos y colores: *varita*: actividades relacionadas con acciones; *huella*: actividades relacionadas con las evaluaciones; *persona*: actividades de otros relacionadas con (*sub-*) *post* propios; *color violeta*: relación de primer grado de la actividad con los *posts*; *color amarillo*: relación de segundo grado de la actividad con los *posts*; *rojo*: relación de segundo grado de la actividad con los *sub-posts*.

Figura 2.17. Esquema de visualización de recompensas en GARUSO (tomada de Kolpondinos y Glinz (2020))

Al final de este proceso se evaluó, entre otras cosas, la aportación de los *stakeholders* al descubrimiento y priorización de las necesidades que debería satisfacer el sistema *SmaWoMo*. En este sentido, la información recopilada por GARUSO permitió obtener los siguientes datos:

- El 37.5% de los *stakeholders* demostró tener un conocimiento competente.
- El 34.3% de los *stakeholders* demostró tener un conocimiento bastante competente.
- El 12.5% de los *stakeholders* demostró tener un conocimiento experto.
- El 6.2% demostró tener un conocimiento avanzado.
- Por otro lado, únicamente el 3.12% demostró no tener conocimiento competente sobre las necesidades que deberían cubrirse con el sistema y el 6.25% restante no reveló ninguna información al respecto.

Los resultados de la evaluación empírica evidenciaron que GARUSO funcionó correctamente. Esta afirmación se argumentó considerando el hecho de que la plataforma logró involucrar a *stakeholders* de diferentes países, lo que resultó en la identificación de un total de 63 participantes fuera del alcance de la organización. Más de la mitad de éstos contribuyeron activamente en la elicitación de las necesidades para el sistema *SmaWoMo*. La evaluación también proporcionó evidencia de que la gamificación se puede aplicar de manera efectiva para motivar a los *stakeholders* a participar en procesos colaborativos de ER. Además, los resultados obtenidos también destacaron la importancia de la motivación personalizada a través de las recompensas como base para la obtención de mejores resultados a través de la aplicación de los principios de la gamificación. Finalmente, se derivó un conjunto de principios de diseño que pueden ser de utilidad para otros interesado(a)s en identificar y motivar a los *stakeholders* que están fuera del alcance de la organización en el contexto de la ER.

2.4.5.2. *Requengin*: Un juego serio para enseñar los fundamentos del estándar ISO/IEC/IEEE 29148

2.4.5.2.1. Objetivo

De acuerdo con García et al. (2020), los JS son una excelente opción educativa para estudiantes de licenciatura, ya que los ayuda a adquirir y refinar conocimientos teóricos al experimentar y aprender de sus propios errores en, por ejemplo, la IR. Lamentablemente, existen varios problemas relacionados con la enseñanza de estándares relacionados con esta área de la IS, entre los que se encuentran los siguientes: se trata de documentos que utilizan un lenguaje demasiado técnico que a menudo es imposible entender a nivel licenciatura, la mayoría de lo(a)s profesores desconocen cómo debe analizarse y aplicarse un estándar y, aunque lo entendieran, el aprendizaje de los estándares del proceso de *software* no es un tópico incluido en los planes de estudio. Considerando este escenario, *Requengin* fue creado como una herramienta de soporte para cursos sobre IR a nivel licenciatura que pretendan enseñar a lo(a)s estudiantes los fundamentos de esta área desde la perspectiva del estándar ISO/IEC/IEEE 29148:2011.

2.4.5.2.2. Descripción

Como se mencionó anteriormente, *Requengin* estableció un escenario para que lo(a)s estudiantes aprendieran los fundamentos del estándar ISO/IEC/IEEE 29148:2011 (ver Figura 2.18). Con este objetivo en mente, dicha propuesta se diseñó como un juego de exploración de mundo abierto, en donde los alumnos deberán explorar tres escenarios diferentes con el fin de relacionarse con los dos procesos de la IR descritos por el estándar (i.e., definición de los requisitos proporcionados por los *stakeholders* y el análisis de dichos requisitos). Estos escenarios son la casa del avatar escogido por el(la) jugador(a), los pasillos de la biblioteca universitaria, y la biblioteca en sí. Por lo tanto, se pretendió que lo(a)s estudiantes mejorarán su comprensión y aplicación de estos dos procesos mientras exploran la biblioteca con el fin de obtener un conjunto de requisitos, necesarios para cambiar el sistema de gestión tradicional de la biblioteca de su universidad por otro moderno que utilice un *software* como principal medio de mejora. En este sentido, *Requengin* implementó seis funcionalidades principales:



Figura 2.18. Pantalla de inicio de *Requengin* (tomada de García et al., (2020))

- Permite la identificación de los *stakeholders* a través del interés que muestran los *stakeholders* en el desarrollo del nuevo sistema de gestión. Dicha identificación se realiza cuando lo(a)s

estudiantes evalúan la relevancia, conocimientos, habilidades y actitudes de los NPC del juego hacia el proceso. Sin embargo, es importante mencionar que el juego no implementó alguna metodología o principio lógico que le indique a lo(a)s estudiantes cómo identificar correctamente a estos NPC.

- Proporciona orientación a lo(a)s estudiantes para la aplicación correcta de la técnica de entrevistas con el objetivo de obtener las necesidades de los *stakeholders* con respecto al nuevo sistema. El juego contiene un conjunto de preguntas y respuestas, las cuales deberán ser seleccionadas por los estudiantes. En ocasiones, el juego podrá detectar patrones de preguntas que pueden desencadenar nuevas respuestas, cuando esto ocurre se le envía una notificación al estudiante, lo que significará un aumento en su puntuación (ver Figura 2.19).
- Proporciona una guía que le permite a lo(a)s estudiantes definir restricciones, identificar servicios, establecer interacciones entre usuarios y el sistema, e identificar requisitos críticos. Lo(a)s estudiantes deben revisar la lista de requisitos con el fin de identificar actores, restricciones o decisiones gerenciales.
- Proporciona una guía para que lo(a)s estudiantes aprendan a analizar y mantener los requisitos de los *stakeholders*. Lo(a)s estudiantes registran los requisitos con el fin de crear una línea base la cual pueda ser analizada y modificada posteriormente.
- Capacita a lo(a)s estudiantes para que puedan definir correctamente los requisitos. Lo(a)s estudiantes utilizan una lista de necesidades de los *stakeholders* que les facilita la identificación de requisitos y la posterior obtención de casos de usos, con el fin de mejorar su conocimiento con respecto a los actores y su interacción considerando la propuesta de un nuevo sistema basado en *software*.
- Proporciona soporte a lo(a)s estudiantes para analizar y mantener los requisitos. Lo(a)s estudiantes deben analizar los requisitos funcionales y no funcionales y generar casos de uso a partir de éstos (ver Figura 2.20). Una vez terminado todo el proceso dentro del juego, lo(a)s estudiantes enviarán la información generada a sus profesores con el fin de recibir realimentación sobre su desempeño en el juego.



Figura 2.19. Uso de la técnica de entrevistas en *Requengin* (tomada de García et al., (2020))



Figura 2.20. Generación de casos de uso en *Requengin* (tomada de García et al., (2020))

2.4.5.2.3. Resultados

Se realizó una evaluación empírica sobre *Requengin* que incluyó la participación de 80 estudiantes del tercer año de la carrera de Licenciatura en Informática de una universidad pública de México. Adicionalmente, cuatro profesores con más de 12 años de experiencia en la enseñanza de la IS a nivel licenciatura observaron a los estudiantes durante la evaluación, brindaron realimentación y orientación, y analizaron su desempeño durante las sesiones de juego. La participación de lo(a)s estudiantes en esta evaluación empírica no influyó en la evaluación general del curso, con el fin de no presionarlo(a)s a participar o que sintieran la necesidad de lograr una puntuación alta.

En este sentido, lo(a)s estudiantes recibieron clases teóricas sobre los procesos de la IR, técnicas empleadas en las actividades definidas por estos procesos, y fundamentos de la norma ISO/IEC/IEEE 29148:2011, para posteriormente participar en sesiones de juego con *Requengin* con el objetivo de poner en práctica los conceptos aprendidos en el aula. Durante las sesiones de juego, los cuatro profesores monitorearon y controlaron el trabajo de los estudiantes con el objetivo de determinar su nivel de comprensión de los procesos y actividades del estándar, y el uso adecuado de las técnicas de la IR. Finalmente, se recogieron las percepciones de los estudiantes mediante la aplicación de dos cuestionarios al final de las sesiones de juego con el fin de evaluar su motivación, sus experiencias con el juego, sus resultados de aprendizaje y sus experiencias en relación con el enfoque de aprendizaje.

La evaluación hizo uso del Modelo de Evaluación de Juegos Educativos (MEEGA, por sus siglas en inglés) para recoger las opiniones de lo(a)s estudiantes participantes sobre *Requengin*. Este modelo permitió evaluar tres categorías (i.e., la motivación, experiencia y resultados de aprendizaje de lo(a)s estudiantes) y once dimensiones del juego para determinar la efectividad educativa durante las sesiones de juego. El cuestionario de MEEGA consta de 27 ítems y se basa en una escala tipo *Likert* de 5 puntos, donde 2 significa que se está muy de acuerdo y -2 muy en desacuerdo con cada ítem. En el contexto de la evaluación empírica realizada, se realizó una personalización a esta escala mediante las siguientes alternativas de respuesta: “Totalmente en desacuerdo” (TD= -2), “En desacuerdo” (ED= -1), “Neutral” (N= 0), “De acuerdo ” (DA= 1), y “Totalmente de acuerdo” (TA= 2), las cuales fueron convertidas a un valor numérico para obtener una medida cuantitativa. En cuanto a la experiencia de aprendizaje, se diseñó un cuestionario adicional de seis ítems que también fue

aplicado tras finalizar las sesiones de juego con *Requengin* para obtener las respuestas de lo(a)s estudiantes.

Finalmente, con el objetivo de aumentar la validez de la evaluación empírica sobre *Requengin*, se diseñó y aplicó un cuestionario final de ocho ítems para recopilar las opiniones de los cuatro profesores participantes.

Los resultados de la evaluación condujeron a la obtención de las siguientes conclusiones:

- El 44% de lo(a)s participantes afirmó estar “Totalmente de acuerdo” en que aprendió a identificar correctamente a los *stakeholders* jugando *Requengin*.
- El 61.6% argumentó estar “Totalmente de acuerdo” en que el juego lo(a)s ayudó a comprender cómo realizar una entrevista productiva para obtener las necesidades y deseos de los *stakeholders*.
- El 100% estuvo “Totalmente de acuerdo” en que durante las sesiones de juego aprendió a definir las necesidades de los *stakeholders* y los requisitos del sistema mediante la aplicación de técnicas populares de la IR.
- El 54.4% indicó estar “Totalmente de acuerdo” en que el juego les permitió aprender a analizar las necesidades de los *stakeholders* y los requisitos del sistema mediante la evaluación de las características definidas por el expertos.
- El 62.4% afirmó que aprendió a definir correctamente los requisitos funcionales y no funcionales dentro del juego.
- Por último, el 100% de lo(a)s participantes consideró estar “Totalmente de acuerdo” en que *Requengin* podría ser una herramienta educativa eficiente para enseñar/aprender los fundamentos del estándar ISO/IEC/IEEE 29148:2011.

2.5. Consideraciones finales sobre el estado del arte

A pesar de que GARUSO y *Requengin* son propuestas cuyo objetivo es la identificación de los *stakeholders*, cada una plantea un enfoque diferente al propuesto en este tema de Tesis. Es decir, GARUSO se enfoca en identificar indiscriminadamente la mayor cantidad de *stakeholders* posibles y, una vez que alcanza un número determinado, los descarta dependiendo su nivel de motivación hasta obtener un conjunto definitivo de *stakeholders*, algo que lamentablemente no se puede llevar a cabo en todo tipo de proyecto. Por otro lado, *Requengin* es un JS que, en efecto, aborda los procesos de la IR de acuerdo con el estándar ISO/IEC/IEEE 29148:2011; sin embargo, le presta mayor atención a la actividad de elicitación de requisitos, asumiendo que la identificación de los *stakeholders* se puede realizar considerando únicamente los conocimientos y habilidades de los *stakeholders* con respecto al proyecto que se pretenda realizar. Algo que tampoco es correcto. Por lo tanto, es verdad que ambas propuestas son relevantes para el desarrollo de la presente Tesis, pero la contribución que se realiza con la creación de un JS diferente se enfoca en proporcionar, a estudiantes de nivel licenciatura, un complemento para la enseñanza/aprendizaje específicamente de la identificación de los *stakeholders*.

3. Metodología

Como se mostró anteriormente, el uso de los JS puede incrementar la motivación y alentar a lo(a)s estudiantes a que participen de manera más efectiva y productiva en la IR, lo que permitiría llevar a cabo la elicitación de requisitos claros y completos, ya que mientras más se involucren los *stakeholders* en el proceso de desarrollo, menor será la probabilidad de tener requisitos erróneos. Además, al hacer uso de los JS se disminuye la complejidad del proceso de aprendizaje sobre el desarrollo de *software*, ya que se les proporciona a los estudiantes un espacio interactivo y agradable para aprender. De esta forma es posible afirmar que el uso de un JS puede ser utilizado como una opción óptima no solo en la industria de desarrollo de *software*, sino también en la educación (Gul et al., 2021). Considerando lo anterior, en este capítulo se presenta el desarrollo de un JS digital para la identificación de los *stakeholders* mediante el uso de la plantilla de *Volere* (Robertson y Robertson, 2012).

En este sentido, el diseño e implementación del JS se basó en el consenso entre la Metodología para el Desarrollo de Juegos Educativos basados en Guiones Interactivos (de Lope et al., 2015) y la Metodología para la Producción de Juegos Serios (Larios et al., 2015) (ver Figura 3.1).

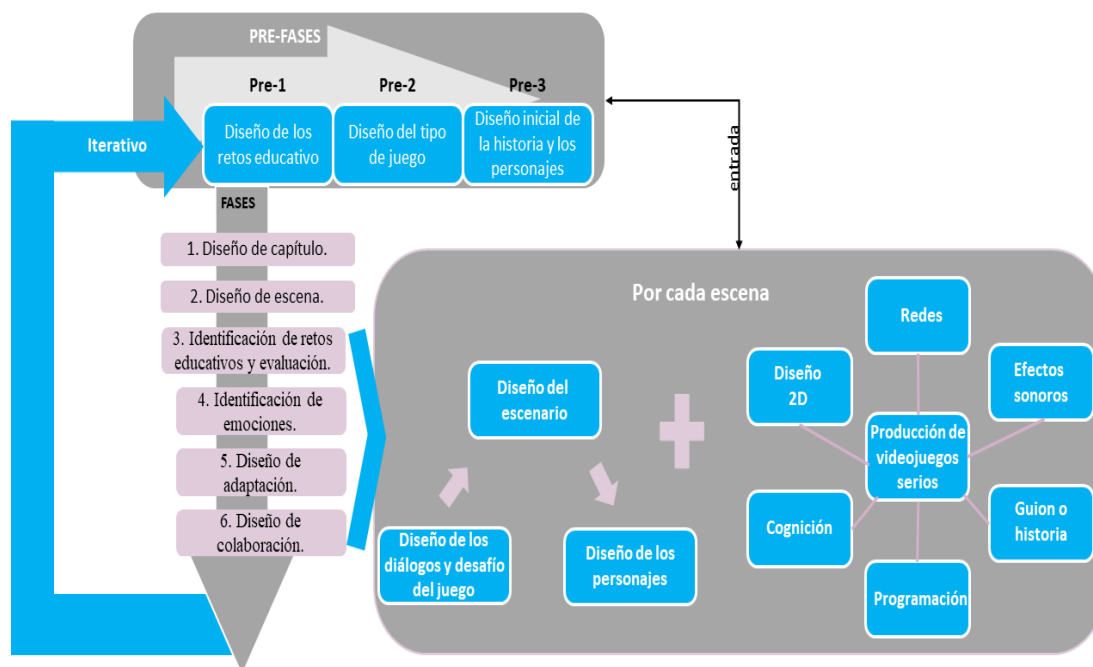


Figura 3.1. Metodología de desarrollo de juegos basada en guiones interactivos y las diferentes grandes áreas de conocimiento comprendidas en el desarrollo de un juego (adaptada de Larios et al., (2015) y de Lope et al., (2015))

Como primer paso para la implementación de esta metodología es necesario implementar tres pre-fases:

- **Pre-1: Diseño de los retos educativos, las competencias básicas y los objetivos educativos.** Como su nombre lo indica, en esta pre-fase los educadores deben establecer los retos, competencias y objetivos que se incluirán dentro del JS, por tal motivo, es común que se realicen entrevistas y se apliquen cuestionarios con el fin de evaluar el resultado final del videojuego, una vez que éste haya sido probado. De igual forma, en el contexto de esta Tesis, es necesario identificar las mejores prácticas para la identificación de los *stakeholders* con el fin de empatarlas con los retos y actividades dentro del JS.
- **Pre-2: Diseño del tipo de juego.** En esta pre-fase se determinan las características fundamentales del JS como el género, el control del avatar, la plataforma, los usuarios futuros, nivel narrativo, área de aplicación y la interactividad del juego.
- **Pre-3: Diseño inicial de la historia y personajes principales.** Para que un juego esté completamente definido y para que, conforme se vaya avanzando en su dinámica, no se pierda continuidad en la historia, es necesario diseñar una historia inicial que la sintetice. Para que esto se pueda cumplir es necesario llevar a cabo las siguientes fases:
 - *Fase 1: Diseño del capítulo.* Un capítulo está definido como un elemento importante en el que se lleva a cabo la organización de la historia y en donde se implementa una forma de facilitar la integración de los contenidos. Es necesario que cada juego implementado tenga por lo menos un capítulo, el cual debe tener coherencia y contar con una secuencia que permita el desarrollo de la historia. Con el fin de proporcionar un orden a los capítulos se establece un diagrama de flujo que describe las acciones que se llevarán a cabo dentro del capítulo, el cual contendrá tanto el nombre del capítulo como las acciones a realizar.
 - *Fase 2: Diseño de la escena.* Cada capítulo del juego está conformado por escenas, las cuales van a determinar la trama, por lo que se deben especificar a través de un diagrama de flujo. Dado que no existe un único flujo, existirán escenas opcionales dependiendo de la decisión del jugador. De acuerdo con Bueno (2018), cada escena debe ser descrita mediante un nombre y un resumen que permitan identificar los siguientes elementos.
 - **Diseño del escenario.** El diseño de un escenario es de gran importancia, ya que éste puede influir en el éxito o fracaso del videojuego. Los escenarios deben ser originales, estar bien diseñados y contar con el mayor número de detalles posibles. Mientras más características de las antes mencionadas tenga, más divertido e innovador será el juego.
 - **Diseño de los personajes.** Éste es esencial cuando se busca crear un videojuego con una historia desarrollada, ya que la participación de un personaje es de suma importancia para que la historia tenga sentido.
 - **Diseño de los diálogos y retos del juego.** Los diálogos dentro de los videojuegos son necesarios para aportar profundidad a la historia. Éstos deben ser cortos, estar bien escritos y ser atractivos para los jugadores, ya que deben ser capaces de transmitir emociones y sensaciones para que éstos se puedan sentir involucrados con la historia.

- *Fase 3: Identificación/etiquetado de los retos educativos y evaluación.* Es necesario que cada vez que un objetivo educativo sea alcanzado, se le haga saber al(a) jugador(a) por medio de una acción o un cuadro de diálogo.
- *Fase 4: Identificación/clasificación de emociones.* Es necesario resaltar las partes del diálogo que van a provocar una reacción emocional en el(la) jugador(a), de tal forma que en el momento de que esto se lleve a cabo, se genere en éste(a) algún tipo de emoción que le incentive un mayor interés por el juego.
- *Fase 5: Diseño de la adaptación.* Es necesario definir atributos que personalicen el juego para así poder determinar si éste logra o no acoplarse a las necesidades y características del(a) jugador(a). Algunos de estos atributos son los retos educativos, el tipo de narrativa, el tipo de interacción, etc.
- *Fase 6: Diseño de la colaboración.* Se deben definir qué desafíos, dentro de las tareas, van a poder realizarse entre grupos de jugadores.

A continuación, se presenta el desarrollo de las fases y pre-fases descritas anteriormente.

3.1. Pre-1: Diseño de los retos educativos, las competencias básicas y los objetivos educativos

Como primer paso se analizaron los conocimientos y habilidades que un ingeniero(a) de *software* debería tener sobre la IR de acuerdo con los Lineamientos Curriculares para Programas de Licenciatura en Ingeniería de *Software*¹⁰ (SE2014, por sus siglas en inglés), el Modelo de Competencias de la Ingeniería de *Software* (SWECOM, por sus siglas en inglés)¹¹ y el Plan de Estudios de Computación 2020 (CC2020, por sus siglas en inglés)¹². Haciendo un análisis de estos tres documentos se obtuvo lo siguiente:

En primera instancia, los estudiantes deben tener claros los conceptos y principios básicos de la IS, las herramientas y las técnicas que se utilizan, así como los métodos que se modelan. Una vez comprendido todo esto, los estudiantes podrán aprender los conceptos de la IR que les permitan llevar a cabo la elicitación, análisis, especificación y gestión de los requisitos, punto de partida de cualquier desarrollo de *software*. Y para esto, deben ser capaces de, en primer lugar, identificar correctamente a los *stakeholders* y tomar en cuenta tanto sus deseos, necesidades y restricciones, para posteriormente traducirlos en requisitos que configuren una solución al problema inicial y presentarlos a los *stakeholders* explicando los posibles impactos que ésta puede ocasionar (sociales, ambientales, etc.).

¹⁰ El SE2014 es un documento creado por la Asociación de Maquinaria Computacional (ACM por sus siglas en inglés) y la Sociedad de Computación del Instituto de Ingeniero(a)s en Eléctricos y Electrónicos (IEEE, por sus siglas en inglés) para profesores universitarios que buscan crear nuevos programas de licenciatura o revisar y actualizar los existentes. Estas pautas también brindan información útil a los empleadores potenciales sobre las habilidades y el conocimiento que se espera de lo(a)s graduado(a)s en cuanto a la IS (SE, 2014).

¹¹ Es un documento creado por el IEEE y la Sociedad de Computación del IEEE para especificar las competencias (i.e., conocimientos y habilidades) de lo(a)s ingeniero(a)s de *software* que participan en el desarrollo y modificación de sistemas intensivos de *software* (SWECOM, 2014)

¹² El CC2020 es una iniciativa de la ACM, IEEE y de la Sociedad de Computación del IEEE lanzada con el fin de resumir y sintetizar el estado actual de los lineamientos educativos de la Computación, así como proponer una visión para futuros lineamientos curriculares (ACM-CC, 2020).

Considerando lo anteriormente mencionado, se busca que el estudiante tenga los conocimientos sobre:

1. Identificación de todas las posibles fuentes de requisitos (*stakeholders*).
2. Identificar clases de usuarios y sus características, de acuerdo con la plantilla de *Volere*.
3. Identificar y consultar con los actores del sistema.
4. Utilizar esquemas para caracterizar y evaluar las relaciones apropiadas entre todos los *stakeholders*.

Posteriormente, para establecer las competencias básicas y los objetivos educativos del JS, se analizaron investigaciones relacionadas con el establecimiento de actividades para facilitar la identificación a los *stakeholders*. En este sentido, en la Revisión Sistemática de Literatura (RSL) realizada por Pacheco y García (2012), se encontraron cuatro prácticas efectivas dentro de la **elicitación de requisitos** para llevar a cabo la identificación de los *stakeholders*, las cuales se resumen a continuación:

- **Identificar y consultar todas las posibles fuentes de requisitos.** Los *stakeholders* deben cumplir con las demandas en términos de experiencia y conocimientos necesarios sobre el dominio del problema. Los líderes de proyectos deben estar capacitados y experimentados en cuanto a la IR. Por lo tanto, se pensó que el contexto de un JS, los *stakeholders* serían únicamente las personas que de cierta forma estuvieran involucradas en el atentado donde perdió la vida una víctima que viajaba en un vagón de tren (i.e., policía encargado del caso, alcalde de *Wonderland*, experta forense, esposa de la víctima, la testigo, secretaria del alcalde, periódicos del día del accidente y cinco días anteriores, cinta de video del día del atentado y un fisgón), debido a que son las que tendrían el conocimiento necesario sobre lo que ocurrió y podrían considerarse sospechosos, ya que pueden proporcionar información sobre los eventos del fatídico día. De igual forma, los documentos y la cámara de seguridad, que forman parte también de los *stakeholders*, proporcionarían información sobre quién puede ser el culpable o quién puede proporcionar pistas para resolver el caso.
- **Identificar clases de usuarios y sus características.** Es importante tener en cuenta que cada uno de los *stakeholders* involucrados desempeña un rol dentro de un proyecto, cada rol tiene diferentes deseos y necesidades y cuenta con un nivel de prioridad que es necesario tomar en cuenta al momento de su identificación. En el caso del JS, el(la) detective, al momento de interrogar a los sospechosos, debería tomar en cuenta su nivel de involucramiento en el asesinato y, en base a esto, debería formular preguntas que le permitan obtener la mayor información posible para resolver el caso. En este sentido, a continuación se presentan de manera muy concisa las características de los personajes que fueron considerados en el escenario que se planteó en el JS:
 - **Detective.** Este personaje deberá obtener la información sobre el asesinato en la estación de trenes, así como identificar posibles sospechosos. En este sentido, él deberá interrogarlos para tratar de obtener la mayor información posible que conlleve la pronta resolución del caso.
 - Esposa de la víctima. Este personaje tiene relevancia al inicio del juego ya que le proporcionará al(a) detective la agenda de la víctima, que contiene información sobre los posibles sospechosos. Sin embargo, el(la) detective no le podrá dar detalles del caso.

- **Policía encargado del caso.** Este personaje tendrá la información inicial del caso (inspección del lugar del asesinato, fotos del lugar y de la situación en que se encontró a la víctima, información de la autopsia de la víctima, y el primer testimonio de la testigo) que le tendrá que comunicar al(a) detective. Así mismo podrá tener información que el(la) detective le vaya proporcionando sobre las actualizaciones en el caso o las pistas.
- **Alcalde de *Wonderland*.** Al ser la persona con mayor jerarquía en el juego y quién se juega su puesto si no se resuelve el asesinato, al alcalde se le podrá dar información sobre el caso y sus avances, esto con el fin de que él haga todo lo posible para que este hecho no se vuelva a repetir en la ciudad.
- **Experta forense.** Este personaje deberá proporcionar al(a) detective toda la información que obtenga del análisis del cuerpo de la víctima y que ayude en la resolución del caso. Sin embargo, no podrá ser informada sobre las actualizaciones del caso.
- **Testigo.** Este personaje tendrá información superficial del caso, ya que a pesar de que durante el asesinato viajaba con la víctima y fue el único que vio al culpable, perdió el conocimiento poco después del atentado, por lo que no tiene una noción clara de cómo se desarrolló éste, pero proporcionará las características físicas del culpable del asesinato.
- **Secretaria del alcalde.** Este personaje al igual que a la testigo, solo se le podrá proporcionar información superflua cuando se requiera una cita con el alcalde para informarle de algún avance en la investigación. Es importante no revelar información extra del caso ya que, al ser la culpable, podría hacer algo para escapar o pagarle al fisgón para que proporcione información errónea.
- **Fisgón.** Este personaje proporcionará información irrelevante del asesinato, pues solo le interesa que le den dinero por dicha información errónea. Es importante no darle ninguna pista del caso, ya que puede venderla al mejor postor (incluido el culpable).
- **NPC.** Son personajes de relleno que podrán ser visualizados alrededor del juego y que no tienen relevancia alguna ya que no poseen ninguna información del caso (i.e., el jefe de la víctima, un compañero de trabajo, un vecino conflictivo (conductor del tren) y un amigo).
- **Identificar y consultar con los actores del sistema.** Se deben seguir pautas que aseguren que solo se identifiquen los *stakeholders* apropiados dentro de cada categoría de las clases propuestas por la plantilla de *Volere* (Robertson y Robertson, 2012). En este sentido, la Tabla 1 muestra el mapeo entre los personajes propuesto para el JS y la plantilla antes mencionada.
- **Utilizar esquemas para caracterizar y evaluar las relaciones apropiadas entre todos los *stakeholders*.** Se pueden usar las siguientes preguntas de investigación: ¿Qué poder tiene esta persona sobre el proyecto?, ¿Qué persona está a cargo de?, ¿Esta persona es asistente de?, ¿Qué tan experto(a)s son en lo que hacen?, ¿Él/ella es crucial para?, ¿Él/ella proporciona información para?, ¿Cuál es su disposición para comunicarse o negociar?, ¿Cuál es su relación con los demás *stakeholders* (en términos de jerarquía, amistad o enemistad)? Es importante tener en cuenta que el jugador(a) debe hacer uso de estos esquemas para poder ubicar correctamente a los *stakeholders* involucrados en el JS ya que su uso garantiza una

identificación menos propensa a equivocaciones. Por otro lado, en caso de que no se usen tales esquemas, el(la) jugador(a) recibirá penalizaciones (e.g., perder dinero, perder pistas, etc.).

Tabla 1. Mapeo de personajes del juego con las clases de *Volere*

Personajes	Clases
Detective	Operador normal.
Policía	Miembro del equipo central (<i>Core team member</i>).
Alcalde	Patrocinador.
Experta forense	Miembro del equipo central.
Testigo	Cliente (<i>customer</i>).
Fisgón	<i>Stakeholder</i> negativo.
Informante anónimo	Soporte operacional.
Periódicos	Consultor externo.
Cámaras de video	Tecnología de Interfaz.

Así mismo, en la misma RSL, Pacheco y García (2012) encontraron que existen clases de *stakeholders* que siempre están presentes, sin importar el tipo de proyecto a desarrollar: equipo de desarrollo, clientes, patrocinadores, operadores normales y *stakeholders* negativos (ver Figura 3.2).



Figura 3.2. Clases de *stakeholders* (traducida de Robertson y Robertson (2012))

3.2. Pre-2: Diseño del tipo de juego

El juego propuesto en esta Tesis recibe el nombre de “*Misterio en Wonderland*”. La historia se desarrolla dentro de la ciudad de *Wonderland*, famosa por tener la estación de trenes más grande de todo el país. Como primer paso dentro del juego, el(la) estudiante debe escoger un avatar (i.e., hombre,

mujer, no binario), el cual representará a un(a) detective dentro del juego y con el cual se podrá interactuar con el fin de identificar a los *stakeholders*. De esta forma, el estudiante podrá aprender y aplicar correctamente la identificación de los *stakeholders* sin saber que está aprendiendo.

3.2.1. Género

El género del JS se relaciona con dos diferentes categorías: la primera, de aventura, que corresponde a la mayor parte del juego y, la segunda, de acción, cuya presencia no es tan notoria dentro del JS; sin embargo, es importante para el desarrollo de la trama. Al incluir estas dos categorías, se busca que el JS logre acoplarse a los géneros que son tendencia en la actualidad, haciendo más amena y divertida la jugabilidad. Esto último se logra al no enfocar el JS en situaciones comunes y conocidas en el mundo real.

3.2.2. Plataforma

El JS está disponible únicamente para computadoras (de escritorio o portátiles) que cuenten con sistemas operativos Windows® 8, 10 y 11.

3.2.3. Propósito y público objetivo

El propósito del JS, como se mencionó anteriormente, es mejorar el proceso de enseñanza/aprendizaje de la identificación de los *stakeholders* en estudiantes de nivel licenciatura con ayuda de la plantilla de *Volere* (Robertson y Robertson, 2012). Aunado a esto, el JS representa una forma amena de aprendizaje.

Teniendo esto en cuenta, el JS está dirigido a estudiantes de licenciatura que cursen la carrera de Ingeniería en Computación, o carreras afines, y que hayan cursado o que se encuentren cursando algún curso relacionado con la IS.

3.2.4. Área de la aplicación

Este JS tomó como enfoque único la identificación de los *stakeholders*, actividad inicial de la elicitación de requisitos (i.e., primera fase dentro de la IR), debido a su importancia dentro del proceso de desarrollo de *software*. Además, son muy pocos los juegos que se desarrollan para cubrir esta fase en un curso tradicional sobre IR (ver Sección 2.4.3).

3.3. Pre-3: Diseño inicial de la historia y personajes principales

El JS comienza explicando un atentado que ocurrió en la estación de trenes de la ciudad de *Wonderland* y que, según la evidencia obtenida por el policía de la ciudad, apunta a que fue provocado por un villano(a) que ha cometido otros atentados en otros puntos de la ciudad desde 5 días anteriores a éste y que, desafortunadamente, ha cobrado su primera vida. Por tal motivo, el alcalde de la ciudad busca al(a) detective más famoso del mundo de nombre *Phoenix*, y que nunca ha perdido un caso, con el fin de que resuelva el misterio y encuentre al culpable.

De esta forma, el(la) estudiante o jugador(a), que en el juego desempeñará el rol del(a) detective, empieza su camino recorriendo toda la ciudad de *Wonderland* e interactuando con todos los *stakeholders* del caso: alcalde de *Wonderland*, policía asignado al caso, esposa de la víctima, agenda de la víctima, experta forense, testigo, periódicos de la ciudad, cinta de video, fisgón, informante anónimo, secretaria particular del alcalde y NPC's. Sin embargo, estas interacciones y los recorridos no serán tarea fácil, ya que algunos implicados en el caso le pedirán al(a) jugador(a) que realice actividades extra o que pague dinero para que le brinden información. De igual forma, rumbo al

desenlace del juego, el(la) jugadora será perseguido(a) por los secuaces del culpable, haciendo más difícil llegar con vida al final del juego.

En este sentido, cada vez que el jugador(a) interrogue con éxito a un sospechoso(a) recibirá una recompensa económica y un aumento en sus puntos, pero en caso de que no logre obtener datos que lo lleven al culpable, perderá información valiosa lo cual le dificultará encontrar quién es el culpable y obtener las recompensas que se obsequiarán dentro del juego. Estas pueden ser dinero y puntos que, además de ayudarlo al jugador(a) a comprar información y pistas que le faciliten la búsqueda del culpable, también le permitirán comprar ropa y mascotas con las que podrá decorar su avatar.

Cómo último punto, existirán sospechosos que van a ser fáciles de descifrar; sin embargo, si el jugador(a) quiere interrogarlos cuando se le ha asignado como tarea interrogar a otro(a)s, no podrá hacerlo, ya que esto(a)s no estarán habilitados para hablar. Una vez que se les asigne la misión o tarea de interrogarlo(a)s, se habilitarán. Esto con el fin de obtener un orden y un flujo de comunicación controlado.

3.3.1. Flujo del juego

Como primer paso, el(la) jugador(a) debe iniciar la aplicación del JS, en donde va a visualizar el menú principal de la Figura 3.3. En este menú el(la) jugador(a) puede elegir entre las opciones de “Jugar”, “Acerca de” o “Salir”; sin embargo, como se desea iniciar el juego, el(a) jugador(a) deberá hacer *clic* en “Jugar”.



Figura 3.3. Pantalla de inicio del juego “*Misterio en Wonderland*”

Una vez que el juego haya iniciado, se presentará al jugador(a) un pequeño tutorial para proporcionarle información sobre las herramientas que le permitirán interactuar correctamente en el juego. La información proporcionada será la siguiente:

- Movilidad del jugador. Las teclas A, W, S y D permitirán que el(a) jugador(a) pueda moverse a la izquierda, arriba, abajo e izquierda, respectivamente. Mediante el ratón de la computadora se podrá mover la dirección de la cámara del juego (ver Figura 3.4).

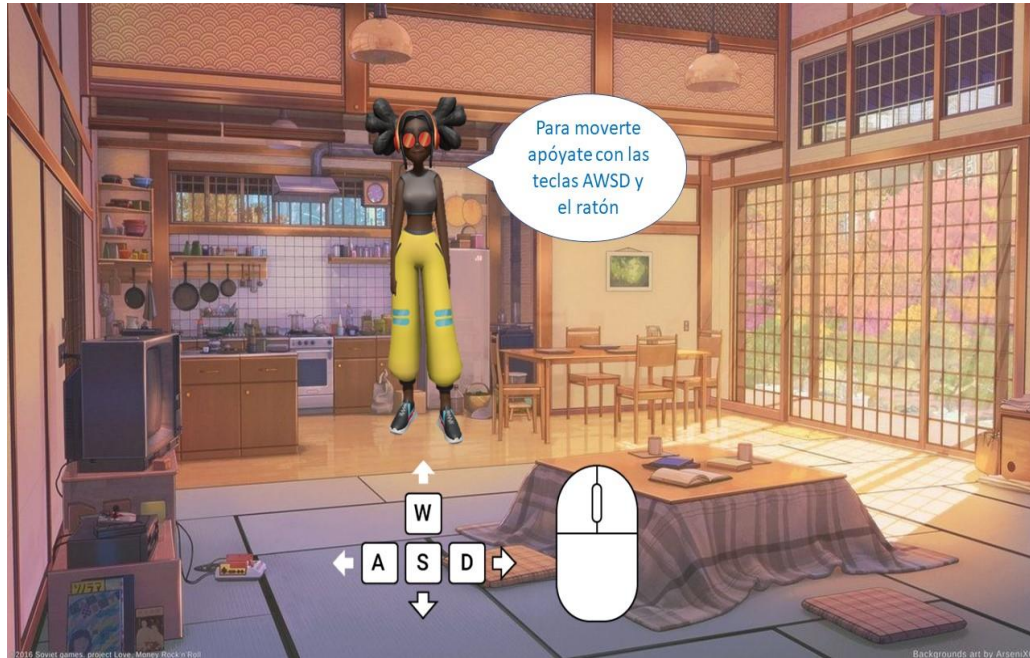


Figura 3.4. Ejemplo del tutorial para explicar la movilidad durante el juego

- Interacción con los objetos y los personajes dentro del juego. Para que el (la) jugador(a) pueda interactuar tanto con los personajes como con los objetos, es necesario oprimir la tecla “Espacio” (ver Figura 3.5).



Figura 3.5. Ejemplo del tutorial para explicar la interacción con personajes y objetos

Como segundo paso, el(la) jugador(a) deberá escoger un avatar, el cual representará al(a) detective dentro del juego y con el cual se deberán hacer los recorridos y tareas establecidas, hasta que se pueda comprar otro si decide cambiarse (ver Figura 3.6).



Figura 3.6. Pantalla para escoger el avatar

El juego inicia con la llegada del(a) detective Phoenix al centro de la ciudad, en donde será recibido por el policía asignado al caso, de nombre René. Una vez juntos, deberán acudir al domicilio de la víctima e informarle a la esposa sobre el atentado en la estación de trenes donde su esposo perdió la vida. En este contexto, el(la) jugador(a) deberá recorrer todo el mapa con el fin de encontrar el domicilio e interactuar con su entorno y familiarizarse con los elementos y personajes que componen el JS (ver Figura 3.7).



Figura 3.7. Ejemplo de interacción del(a) jugador(a) con el entorno

Después de que la conversación haya terminado, el(la) jugador(a) recibirá la agenda de la víctima en la cual podrá encontrar información sobre las personas con las que él ahora occiso se habría reunido recientemente, convirtiéndolos automáticamente en sospechosos. El(la) detective deberá analizar la agenda de la víctima para identificar a cuatro sospechosos: el jefe de la víctima, un compañero de trabajo, un vecino conflictivo (conductor del tren) y un amigo. Sin embargo, la única información que se podrá visualizar para el(a) jugador(a) será el nombre y su relación con la víctima; la imagen y la descripción de cada uno de ellos se desbloquearán cuando el(la) detective los encuentre e interrogue. También deberá buscar los periódicos publicados tanto el día del atentado actual como de los anteriores, con el objetivo de analizarlos y tratar de encontrar información relevante para resolver el caso. Por otro lado, el policía le pedirá al(a) detective que platique con la experta forense sobre las causas de la muerte de la víctima, información que lo(a) ayudará a seguir avanzando en la resolución del caso y que develará que el asesino llevaba un anillo con un símbolo muy particular (una estrella). También le pedirá que revise las cámaras de seguridad de la estación de trenes con el fin de obtener más información. El policía le pedirá al(a) detective que haga todo lo posible por resolver el caso, ya que está en juego la continuación del alcalde en su puesto. Una vez que termine de hablar con el policía, el(la) detective tendrá que entrevistar a los cuatro sospechosos encontrados en la agenda para posteriormente ir a revisar las cámaras de seguridad.

Cuando el(la) detective llegue a la estación de trenes con el objetivo de visualizar las cámaras de seguridad para encontrar al responsable del fatídico atentado, se dará cuenta de que la cinta de la cámara que enfoca el andén donde sucedió el asesinato fue robada (por lo que otra de sus misiones será encontrarla). Sin embargo, viendo a su alrededor, se da cuenta de la presencia de una persona, el fisgón, al cual deberá interrogar y cuestionar por su presencia en ese lugar, ya que solo personal autorizado puede entrar. El fisgón le proporcionará información sobre los rumores que apuntan a que en el círculo cercano del alcalde está el asesino, ya que quieren sabotear su reelección. Es importante tener en cuenta que la cinta podrá ser localizada únicamente cuando el(la) detective haya interrogado a todos los sospechosos y revisado los periódicos del día del accidente y de los atentados anteriores (ver Figura 3.8).

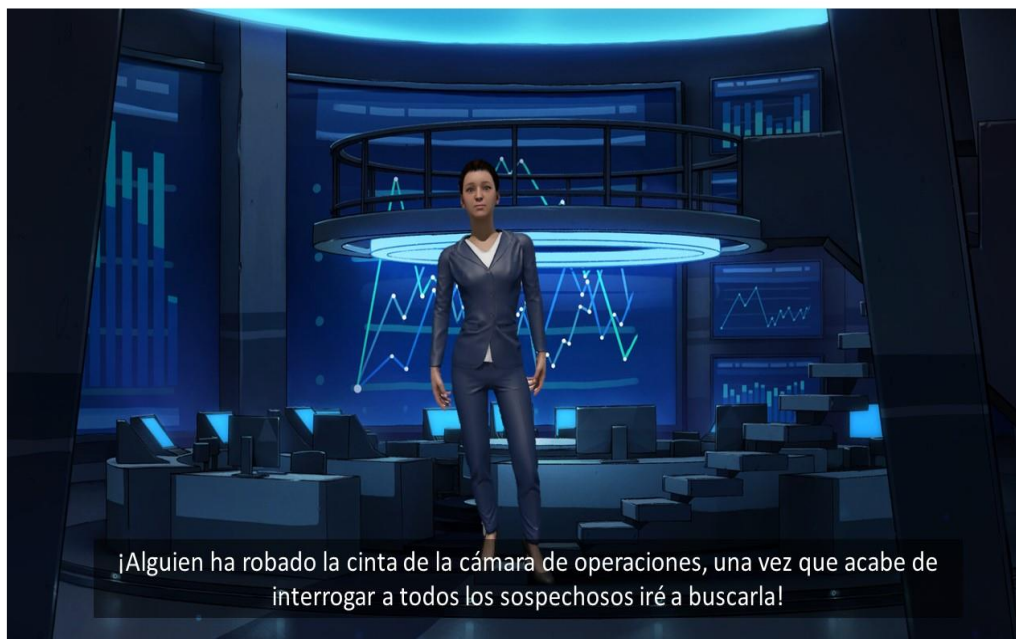


Figura 3.8. Ejemplo de interacción para encontrar el video de la cámara

Una vez que el(la) detective termine de hablar con el fisgón, se le asignará como tarea ubicar e interrogar a la testigo del fatídico accidente – que también era pasajera del tren-, con el fin de obtener información sobre lo ocurrido ese día. En este contexto, el(la) jugador(a) deberá acercarse a la testigo y deberá pulsar la tecla “Espacio” del teclado para comenzar la plática. En caso de que el(la) jugador(a) no inicie una conversación, el juego no podrá continuar y el tiempo seguirá avanzando (ver Figura 3.9).



Figura 3.9. Ejemplo de inicio de plática entre el(la) jugador(a) y algún personaje

Durante el interrogatorio, la testigo le contará al(a) detective que ese día vio a una persona que se le hizo muy sospechosa, ya que estuvo observando el movimiento de pasajeros en el andén pero que nunca lo vio abordar el tren. Sin embargo, debido a la pérdida de conocimiento que sufrió en ese momento, por el impacto del atentado, no recuerda bien ni su vestimenta, ni su cara, más que de forma vaga, pero sí de que llevaba un anillo en la mano derecha, que brillaba mucho. De último momento recuerda haber visto una foto del sospechoso en el periódico del día posterior al accidente en un artículo sobre los atentados acontecidos anteriormente. Por lo que, una vez que el(la) detective ha terminado de interrogar a la testigo debe buscar el periódico indicado.

Cada vez que el(la) detective tenga que realizar una misión, se va a habilitar una ventana donde se mostrará una línea cuya función será la de guiar al(a) jugador(a) a su destino. En el punto final de esta línea (destino) se va a habilitar una flecha y una luz, esto con la intención de que el(la) detective sepa el lugar exacto a dónde llegar para que la misión se cumpla.

Cuando se tenga que tomar un objeto, éste se iluminará de color rojo cuando el(la) detective se encuentre lo suficientemente cerca, lo que le indicará que el objeto puede ser tomado. Una vez que lo haya agarrado, deberá abrir el inventario con el fin de leerlo y analizarlo para obtener información. Por otro lado, y al igual que con las misiones, cada vez que el(la) detective desee interactuar con un sospechoso, se van a habilitar un camino, una flecha y una luz, que se deben seguir (para poder hablar con el sospechoso el(la) detective deberá pararse enfrente de él). Lamentablemente, interrogar a un sospechoso(a) no será tarea sencilla para el(la) detective, ya que algunas veces para acceder a contestar, le pedirán dinero a cambio de la información, o bien, que realice pequeñas misiones e incluso algunas veces deberá someterlos, aunque estas opciones dependerán de cada sospechoso(a).

Es importante tener en cuenta que cada uno de los sospechosos va a proporcionar información sobre dónde encontrar al siguiente (dado que la jugabilidad del JS va a ser lineal). Ejemplo:

- La testigo va a proporcionar información sobre los periódicos, tanto del fatídico día como de los atentados anteriores.
- El fisgón no va a proporcionar ninguna información relevante e interesante.
- Los periódicos, del fatídico día y de los cinco días anteriores, van a proporcionar información sobre atentados similares donde actuó el culpable.

En cuanto a los periódicos, estos contendrán información fotográfica sobre el sospechoso. Por ejemplo: “*El sospechoso del atentado de la estación de trenes fue visto también en el aeropuerto de Wonderland hace cinco días, donde también hubo un accidente que dejó 10 heridos. Siempre va vestido de forma muy rara: gabardina negra larga, sombrero negro, mascada cubriéndole el rostro y guantes negros de piel, así mismo es notorio que porta un anillo muy brillante en la mano derecha*”.

Una vez que todos los sospechosos hayan sido interrogados y que los periódicos hayan sido revisados, el(la) detective deberá elaborar una lista de los tres principales sospechosos y presentársela al alcalde. Sin embargo, estando en la oficina del alcalde se dará cuenta de que tanto este personaje como su secretaria, llevan en la mano derecha un anillo con una estrella. Motivo por el cual, decide no hablar con el alcalde y regresar a su oficina para indagar sobre él y su secretaria, para posteriormente elaborar una nueva lista de sospechosos. Sin embargo, leyendo nuevamente los periódicos encontrará información que lo llevará a descubrir que dentro del círculo de trabajadores cercanos al alcalde existe cierta resistencia a su reelección y que buscan desacreditarlo ante la opinión pública creando caos en *Wonderland* con los atentados. Justo en ese momento recibirá una llamada de un informante anónimo que le indicará en donde puede estar la cinta de video robada (que será la que revele quién es el verdadero culpable del atentado). Cuando el(la) detective llegue al lugar donde le indicó el informante anónimo, éste lo chantajeará con dinero para que le entregue la cinta. Una vez que el(la) detective tenga en su poder la cinta, deberá regresar a la estación de trenes, con el fin de reproducirla y atrapar al culpable. Sin embargo, deberá ser muy rápido y cuidadoso, ya que el culpable ya está enterado de que el(la) detective recuperó la cinta de video del día del atentado en la estación de trenes y hará todo lo posible por impedir que el(la) detective la vea (será atacado por unos maleantes por lo que tendrá que defenderse).

Una vez que el(la) detective esté en la estación de trenes, deberá reproducir la cinta y descubrir al culpable (que es la secretaria del alcalde), quien debería estar en la nueva lista de los tres principales sospechosos. De no ser así el juego deberá terminar, viéndose al culpable escapándose de *Wonderland*. En caso contrario, el(la) detective confirmará sus sospechas y tendrá que ir a un punto alejado de la ciudad de *Wonderland*, en donde deberá de enfrentarse al culpable con el fin de detenerlo y entregarlo al policía encargado del caso. Por lo que el juego deberá terminar cuando el alcalde de *Wonderland* le entregue un reconocimiento por su valentía.

En cuanto a las recompensas dentro del juego, el(la) jugador(a) iniciará con una cantidad determinada de dinero, la cual le ayudará a comprar pistas o a comprar avatares para posteriormente cambiarlos mientras que el(la) jugador(a) interactúa con el juego. Cada vez que el investigador interroga con éxito a un sospechoso, recibirá dinero con el cual podrá acceder a la tienda y comprar artículos y mascotas para decorar el avatar.

3.3.2. Personajes

3.3.2.1. Detective

Es el(la) protagonista o personaje principal del JS. Su nombre es Phoenix y es el(la) detective más famoso(a) del mundo, ya que ha resuelto todos los casos para los que ha sido contratado(a). En el JS se le asigna la tarea de resolver el caso del fatídico atentado sucedido en la estación de trenes de la ciudad de *Wonderland* que ha cobrado la vida de un ciudadano. En este sentido, el(la) investigador(a), en primera instancia, es un personaje serio que puede ser mujer, hombre o no binario.

3.3.2.2. Policía encargado del caso

Personaje secundario cuyo papel es recibir al(a) detective Phoenix en *Wonderland* y proporcionarle la información disponible sobre lo acontecido en el atentado en la estación de trenes. Este personaje le dará las primeras pistas al(a) detective sobre a quiénes debe interrogar para obtener información y atrapar al culpable. Dentro de este juego, el policía es una mujer que utiliza vestimenta de SWAT. Este personaje es uno de los principales proveedores de información, ya que proporciona el contexto y las primeras y más importantes pistas para comenzar el juego.

3.3.2.3. Alcalde de *Wonderland*

Personaje secundario cuyo papel es ser el patrocinador del caso, el personaje principal trabaja bajo el mando del alcalde, por lo que deberá presentarle algunos informes poco detallados sobre quiénes son los posibles sospechosos.

3.3.2.4. Experta forense

Personaje secundario cuyo papel es averiguar las causas de muerte de la víctima. Una vez que se identifique la causa de muerte, se le va a proporcionar al(a) jugador(a) para ayudarlo(a) a amenazar al fisgón para que proporcione información (el objeto que causó la muerte es algo que él vende), en caso de que no se cuente con suficiente dinero para pagarle.

3.3.2.5. Testigo

Personaje secundario cuyo papel es proporcionar la información primaria del caso (i.e., pistas) al personaje principal para empezar su investigación. Este personaje aparecerá dos veces dentro del juego y cada vez será una persona diferente (el primer papel corresponde a la testigo y el segundo papel corresponde a un NPC dentro del tutorial). La segunda vez que se le entreviste, proporcionará información clave sobre quien es el culpable.

3.3.2.6. Secretaria particular del alcalde

Personaje secundario responsable del atentado. Este personaje es, de acuerdo con las pesquisas, el menos relacionado con el caso. Sin embargo, existe una poderosa razón que la convierte en una sospechosa importante: quiere ser la alcalde en la próxima elección, desacreditando mediante los atentados, la posible reelección del alcalde actual. En caso de no ser descubierta, su próxima víctima sería el alcalde de *Wonderland*.

3.3.2.7. Periódicos del día del accidente y cinco días anteriores

Objetos secundarios cuyo papel es proporcionar información sobre algunos sospechosos, es decir, las personas relacionadas con los atentados aportando su media filiación y los últimos lugares donde fueron vistos. Es necesario comprar el periódico para poder leerlo.

3.3.2.8. Cinta de video del día del atentado

Objeto secundario cuyo objetivo es mostrar quién es el culpable del asesinato en la estación de trenes. Una vez que el(la) detective termine de entrevistar a todos los sospechosos deberá ir en búsqueda de este objeto, el cual dará la solución al caso y por ende pondrá un fin al juego. Es importante obtener este objeto porque sin él el juego no tendría continuidad.

3.3.2.9. Fisgón

Personaje secundario cuyo papel es proporcionar información falsa e irrelevante al personaje principal.

3.3.2.10. Informante anónimo

Personaje secundario cuyo papel es proporcionarle al(a) detective la cinta de video que le permitirá resolver definitivamente el caso.

3.3.2.11. NPC

Personajes secundarios que cumplen la única función de estar de relleno dentro del juego. Estos personajes deben ser entrevistados muy brevemente, en caso contrario, el(la) jugador(a) perderá dinero y puntos.

3.4. Fase 1: Diseño del capítulo

El desarrollo del juego serio se llevó a cabo considerando dos capítulos principales: el tutorial y el juego con la temática principal. El primer capítulo (C1_Tutorial) está diseñado para que el(la) estudiante al seleccionar el botón “Inicio” comprenda a través del personaje de una chica, la dinámica del juego, la visualización de los controles, y la explicación de la función de cada uno de los elementos que aparecen en pantalla (e.g., botones, letreros, etc.).

En el segundo capítulo (C2_Principal) se implementa la historia del juego, por lo que está conformado por todas las escenas que permitirán que el(la) estudiante pueda llevar a cabo una correcta identificación de los *stakeholders*. En este contexto, el(la) jugador(a) tiene que interactuar con los personajes con el fin identificar correctamente quiénes pueden proporcionarle información útil que lo lleve a identificar al culpable, a la par que comprenderá que no todos los *stakeholders* van a proporcionarle información de utilidad para lograr su objetivo, logrando así, el aprendizaje objetivo. En este capítulo se llevan a cabo los siguientes retos educativos:

- Identificar y consultar todas las posibles fuentes de requisitos (*stakeholders*).
- Identificar clases de usuarios y sus características.
- Identificar y consultar con los actores del sistema
- Utilizar esquemas para caracterizar y evaluar las relaciones apropiadas entre todos los *stakeholders*.

A partir de que el(la) jugador(a) entre al capítulo C2_Principal deberá llevar a cabo una serie de misiones con el fin de cumplir su objetivo inicial. Lamentablemente, esto no será una tarea sencilla, ya que, para cumplir exitosamente con cada una de las misiones asignadas, el(la) jugador(a) deberá recorrer los diferentes escenarios y, además, dentro de un tiempo asignado. En caso de no hacerlo de manera satisfactoria, el(la) jugador(a) no podrá obtener recompensas únicas dentro del juego. De igual forma, el(la) jugador(a) debe prestar suma atención al momento de realizar las misiones y entrevistar

a los sospechosos, ya que en caso de que no identifique correctamente al culpable perderá puntos y dinero, pero además no podrá terminar el juego, dado que no se desbloqueará el camino en donde se enfrentará con el enemigo, quedando en vergüenza ante toda la población de *Wonderland*.

3.5. Fase 2: Diseño de la escena

El capítulo C1_Tutorial comprende las siguientes escenas:

- C1_1_Tutorial. En esta escena se introduce al(a) jugador(a) a los controles y objetos básicos del juego, también se proporciona información sobre cómo interactuar con los personajes, objetos y botones dentro del juego.
- C1_2_Escoger_Avatar. En esta escena, el(la) jugador(a) deberá seleccionar el avatar que sea de su agrado y empezar el capítulo C2_Principal.

Mientras que el capítulo C2_Principal comprende las siguientes escenas:

- C2_0_Entrada_Ciudad. Esta escena es el inicio del juego. Es aquí donde al(a) jugador(a) se le pondrá en conocimiento sobre el atentado ocurrido ese mismo día en la estación de trenes. De igual forma, se le asignará su primera misión, la cual va a consistir en entrevistar al policía René para que proporcione instrucciones e información que le ayuden a resolver el caso.
- C2_1_Casa_Víctima. En esta escena el(la) jugador(a) deberá visitar, junto al policía, a la esposa del asesinado. Esto con el fin de informarle lo sucedido y obtener información que pueda ayudar con la resolución del caso. Una vez que se hable con la esposa, ésta les proporcionará información útil acerca de una agenda, donde se encontrará información sobre posibles sospechosos (i.e., *stakeholders*). Por tal motivo, en esta escena el(la) jugador(a) también tendrá que localizar las notas de la agenda y analizarlas. Aquí también será posible encontrar al mejor amigo y al vecino conflictivo de la víctima, por lo que el(la) jugador(a) deberá interrogarlos para definir si son culpables o inocentes.
- C2_2_Morgue. En esta escena es donde se localiza la experta forense. Aquí, se encontrará a la médica analizando el cadáver de la víctima por lo que proporcionará información relevante sobre algunas causas de la muerte y el objeto que llevaba el asesino y que dejó marca en el cuerpo.
- C2_3_Estación_Trenes. En esta escena se localiza a la testigo y también es donde se llevará a cabo el combate final, en caso de identificar correctamente al culpable. Este sitio, al ser un lugar muy concurrido por transeúntes y trenes, se caracteriza por ser un lugar ruidoso. En esta escena, el(la) jugador(a) podrá interactuar con la testigo y en su momento, deberá realizar un combate peligroso con los secuaces del asesino que podría llevarlo(a) a la muerte.
- C2_4_Cuarto_Cámaras. En esta escena es donde se ubican tanto el figgón como los videos que van a mostrar al culpable del fatídico atentado. Este sitio es donde se pueden ver los videos de todas las cámaras que hay dentro de la estación de trenes de la ciudad de *Wonderland*, por lo que, en este contexto, esta escena se caracteriza por ser un lugar silencioso al que no todas las personas tienen acceso. En esta escena el(la) jugador(a) deberá interactuar con el figgón y además deberá de reproducir los videos de las cámaras con el fin de encontrar al culpable.
- C2_5_Parque. Esta escena es donde se localizará tanto al compañero de trabajo de la víctima como el puesto de periódicos de la ciudad de *Wonderland*. Es aquí donde se podrán comprar

los dos periódicos que proporcionarán información importante sobre el sospechoso, por lo que deberán analizarse para armar la lista de sospechosos. Además, en esta escena también se deberá interactuar con el compañero de trabajo de la víctima, con el fin de obtener su declaración y saber si puede ser considerado como un posible sospechoso. Este lugar se caracteriza por ser silencioso, ya que es el área de comida de la ciudad.

- C2_6_Parque de juegos. En esta escena es donde se localizará el alcalde para hablar por primera vez con él. Se caracteriza por desarrollarse en un lugar ruidoso y muy animado.
- C2_7_Límites_Ciudad. Esta escena es donde se localiza el informante anónimo. Aquí se deberá entablar una conversación con él con el fin de obtener información valiosa sobre quién es el culpable. Además, es aquí donde se deberán comprar las cintas faltantes del cuarto de cámaras de la estación de trenes, las cuales contienen imágenes del culpable del atentado. Esta escena, al estar alejada de la ciudad, está caracterizada por desarrollarse en un lugar muy silencioso y con poco flujo de peatones.
- C2_8_Área residencial. En esta escena se localizará a la secretaria y es, también, donde se le entregará al alcalde la nueva lista de sospechosos. Esta escena es importante ya que revela información y pistas vitales sobre quién puede ser el asesino. Así mismo, se caracteriza por ser una zona silenciosa.
- C2_9_Final_Perdiste. Esta escena representa el primer final de la historia donde el(la) jugador(a) perdió el juego ya que no supo identificar correctamente a los *stakeholders* sospechosos del caso. Por lo que se muestra al culpable victorioso y burlándose del(a) jugador(a). Esta escena se caracteriza por ser ruidosa, ya que el culpable se va a reír eufóricamente del(a) detective.
- C2_10_Final_Moriste. Esta escena representa el segundo final de la historia, donde se va a localizar la secretaria y a sus tres secuaces, los cuales se estarán burlando del(a) jugador(a) ya que perdió la batalla contra ellos. Esta escena se caracteriza por ser ruidosa, ya que los secuaces de la secretaria estarán bailando y burlándose del(a) jugador(a).
- C2_11_Final_Ganaste. Esta escena representa el tercer final de la historia, puesto que se van a localizar algunos de los ciudadanos de *Wonderland* que estarán festejando la victoria del(a) detective. Esta escena se caracteriza por ser ruidosa, ya que los ciudadanos van a estar muy felices de que se haya resuelto satisfactoriamente el caso.
- C2_12_Plantilla de Volere. La plantilla de *Volere* (Robertson y Robertson, 2012) no es como tal una escena que se encuentre presente dentro del juego; sin embargo, sí representa un apoyo importante de aprendizaje, por lo que se podrá acceder a ella por medio de un botón o cuando se hable con un *stakeholder* clave.

Considerando lo anterior, el flujo entre las escenas del juego de ambos capítulos se muestra en la Figura 3.10. A continuación, se detalla el diseño de cada escena contenida en cada capítulo del juego.

3.5.1. C1_1_Tutorial

En esta escena se llevará a cabo la introducción al juego principal. Aquí, el(la) jugador(a) deberá aprender los comandos, botones, componentes y objetos con los que deberá interactuar dentro de la partida principal. En el tutorial, el(la) jugador(a) no va a tener la opción de escoger a su personaje para interactuar dentro de la escena, sino que se le asignará un personaje por default.

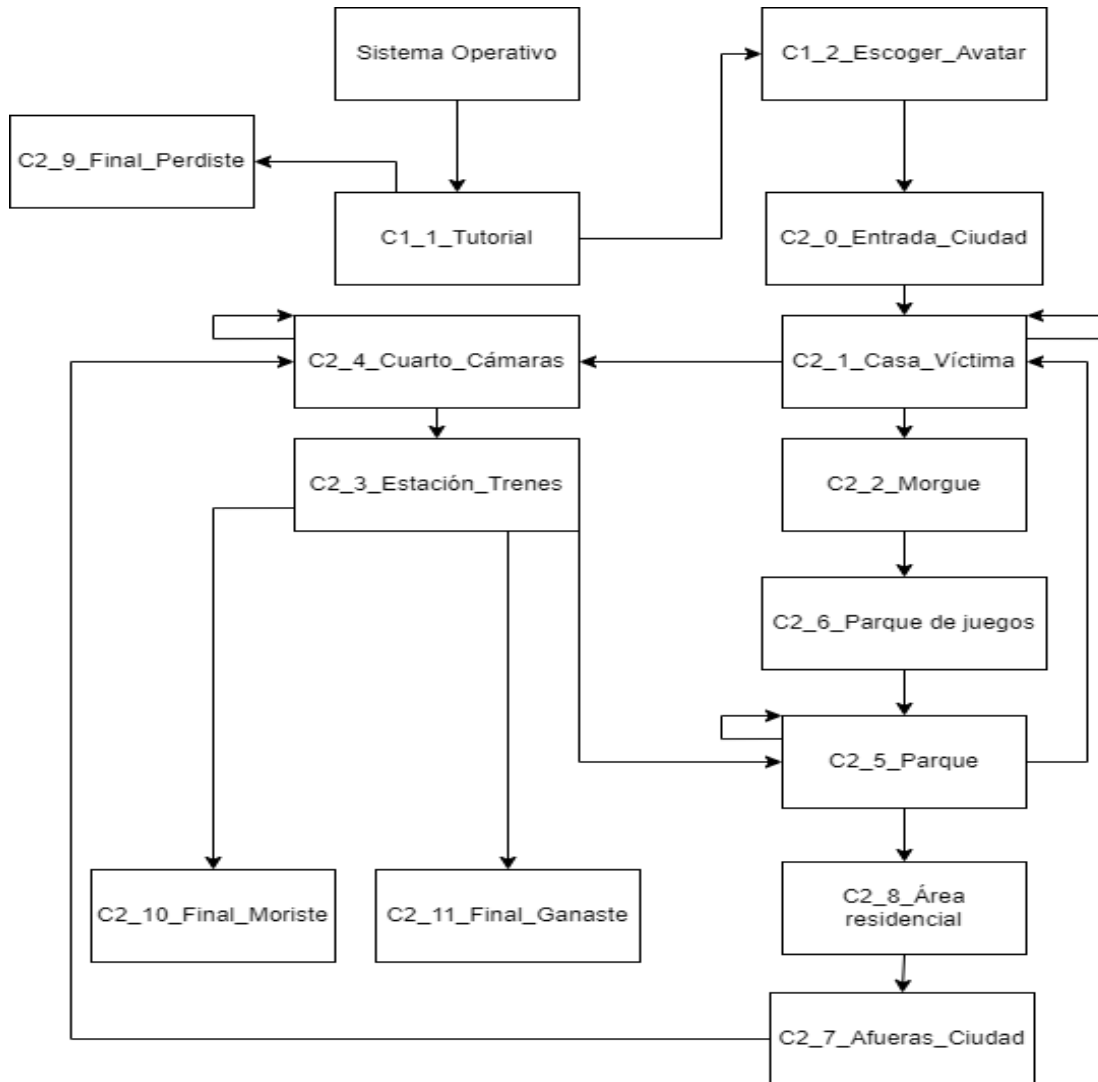


Figura 3.10. Flujo de las escenas entre capítulos del juego

3.5.1.1. Diseño del escenario

Esta escena se desarrolla en un pequeño jardín que incluye los siguientes elementos:

- *Paredes invisibles.* El jardín está rodeado de paredes invisibles las cuales fungen como delimitadores para que el(la) jugador(a) no supere los límites del escenario establecidos dentro del juego.
- *Decoraciones alusivas.* El jardín cuenta con decoraciones como árboles, macetas, plantas, pasto y flores, las cuales permiten darle una representación más real al jardín.
- *Mesas.* El jardín cuenta con tres diferentes mesas, las cuales tienen como objetivo sostener los objetos con los que el(la) jugador(a) va a interactuar.
- *Figuras geométricas 3D* (dos cubos y un cuadrado). Como se mencionó anteriormente, el jardín cuenta con tres diferentes figuras geométricas posicionadas una sobre cada mesa. El objetivo de estos objetos, es ayudar a que el(la) jugador(a) se familiarice con los controles del juego que serán necesarios para interactuar dentro de la partida principal.

- *NPC's*. El jardín cuenta con la presencia de un personaje no jugable, que, al igual que con las figuras geométricas, su función será ayudar al(a) jugador(a) para que se familiarice con los comandos dentro del juego.
- *Botones, textos, imágenes, paneles* (i.e., elementos gráficos de interfaz). A lo largo de esta escena se podrán apreciar en la pantalla los diferentes elementos que serán de ayuda para que el(la) jugador(a) se familiarice con los comandos del juego. Los elementos que aparecerán dentro del tutorial son los siguientes: flechas, un mapa miniatura, un mapa general, un botón de tienda, un botón de resumen de diálogos, un botón de misión actual, un botón de ayuda para identificar a los *stakeholders*, un panel donde se muestra tanto el puntaje como el dinero disponible, un panel donde se indica el tiempo asignado a las misiones, un panel con texto que representa los diálogos, y un panel con texto que despliega las advertencias.

3.5.1.2. Personajes

Los personajes dentro de esta escena son únicamente dos: avatar del(a) jugador(a) y NPC.

3.5.2. C1_2_Escoger_Avatar

En esta escena, el(la) jugador(a) deberá escoger un personaje de tres disponibles (i.e., un hombre, una mujer y un robot) que utilizará a lo largo del juego. Este avatar será reconocido como el personaje principal (i.e., detective) con el cual se deberán identificar y, posteriormente, identificar e entrevistar a los *stakeholders*. Una vez que se haya seleccionado al personaje, se deberá hacer *clic* sobre éste y después sobre el botón “Jugar” para comenzar con la partida principal.

3.5.2.1. Diseño del escenario

Esta escena incluye los siguientes elementos:

- *Avatares*. Esta escena está compuesta por las tres opciones disponibles para el personaje: un hombre, una mujer y un robot.
- *Elementos gráficos de la interfaz*. La escena está compuesta por varios componentes como el botón “Escoja un personaje” que se encuentra ubicado sobre los avatares - cuya función es seleccionar a alguno de ellos cuando se le dé *clic*-, y el texto “Jugar” - que proporcionará una instrucción sobre lo que se debe hacer dentro del juego.

3.5.2.2. Personajes

Los personajes presentes dentro de esta escena son los siguientes: avatar correspondiente a la mujer, avatar correspondiente al hombre, y avatar correspondiente al robot.

3.5.3. C2_0_Entrada_Ciudad

3.5.3.1. Diseño del escenario

Con esta escena se da inicio al juego y es aquí donde el(la) jugador(a) va a parecer por primera vez dentro de la escena principal. Como punto número uno dentro de esta escena, un representante del gobierno de *Wonderland* va a recibir al(a) detective y lo pondrá en contexto sobre el caso asignado. Una vez que termine la conversación, se le asignará la primera misión al (a) jugador(a), la cual será hablar con el policía, el cual le proporcionará información importante para comenzar la resolución del caso.

Esta escena incluye los siguientes elementos:

- *Coche de policía.* Un automóvil tipo patrulla que tendrá la torreta encendida puesto que acaban de recibir la petición de cercar la zona del asesinato.
- *Calle principal.* Es la entrada principal a la ciudad de *Wonderland* y consta de la carretera, la banqueta y el pasto.

3.5.3.2. Personajes

Los personajes presentes dentro de esta escena son los siguientes: policía, NPC Ingeniero industrial y NPC Motociclista.

3.5.4. C2_1_Casa_Víctima

En esta escena, el(la) jugador(a) deberá llevar a cabo seis diferentes misiones, cuatro de ellas se llevarán a cabo dentro de la casa y las dos restantes fuera de la misma. En cuanto a las misiones que se realizarán dentro de la casa, el(la) jugador(a) deberá, como primer paso, llegar al lugar, para posteriormente interactuar con la esposa de la víctima, encontrar las notas correspondientes a la agenda (las cuales estarán esparcidas por toda la casa) y leer dichas notas. Por otro lado, en cuanto a las dos misiones restantes, el(la) jugador(a) deberá interactuar con el vecino conflictivo y el mejor amigo de la víctima.

3.5.4.1. Diseño del escenario

Esta escena incluye los siguientes elementos:

- *Casa principal.* La casa principal es aquella en donde se llevarán a cabo las cuatro primeras misiones y está conformada por:
 - *Sala.* Contiene un sillón, un espejo, decoraciones varias y un librero. Aquí se podrá encontrar una nota.
 - *Cocina.* Contiene una estufa, un lavabo, un refrigerador, una mesa, cuatro sillas, una decoración. Aquí se podrá encontrar una nota más.
 - *Recámara.* Contiene una cama, un buró, un escritorio, una computadora, una silla de escritorio, y decoraciones varias. Aquí se podrá encontrar otra nota.
 - *Baño.* Contiene una bañera, una taza de baño, un lavabo, un mueble para el lavabo, un espejo, un dispensador de jabón, un secador de manos, y un vaso con cepillos de dientes. Aquí también se podrá encontrar una nota.
- *Alrededores de la casa.* Los alrededores de la casa corresponden a la casa de los vecinos, aquí se llevarán a cabo las últimas dos misiones de las seis mencionadas anteriormente. Es importante tener en cuenta que las misiones no se llevan a cabo en un orden consecutivo. Los alrededores están conformados por:
 - *Casa del vecino conflictivo.* Es en esta casa donde se puede interactuar con el vecino conflictivo.
 - *Casas de los vecinos de los alrededores.* Es entre estas casas que se podrá interactuar con el mejor amigo de la víctima.

3.5.4.2. Personajes

Los personajes presentes dentro de esta escena son los siguientes: esposa de la víctima, policía, vecino conflictivo, mejor amigo de la víctima, NPC, y borrego.

3.5.5. C2_2_Morgue

En esta escena el(la) jugador(a) debe realizar dos misiones, la primera consiste únicamente en llegar, precisamente, a la morgue. La segunda misión consiste en interactuar con la experta forense, donde el(la) jugador(a) deberá prestar suma atención para que posteriormente, recuerde la conversación, ya que durante ésta se le dará información vital que lo(a) ayudará a resolver el caso. Es importante tener en cuenta que en el caso de que el(la) jugador(a) no recuerde dicha conversación, podrá apoyarse en el botón “Resumen de diálogos”, el cual estará disponible durante todo el juego. La información dentro de este panel se va a desbloquear cada vez que el(la) jugador(a) hable con los *stakeholders* clave para la resolución del caso.

3.5.5.1. Diseño del escenario

Esta escena incluye los siguientes elementos:

- *Escena del crimen.* Contiene cinco conos, cinta policiaca, el arma de fuego con la que la víctima fue ultimada, las balas del arma de fuego, y una sábana manchada de sangre con la que se cubre al cadáver.

3.5.5.2. Personajes

Los personajes presentes dentro de esta escena son los siguientes: policía, experta forense, y cadáver de la víctima.

3.5.6. C2_3_Estación_Trenes

Esta escena es importante para el desarrollo del juego, ya que aquí se llevarán a cabo cinco misiones decisivas: a) Dirígete hacia la dirección de la testigo y habla con ella, b) Ve a la estación de trenes, c) Agarra la espada que está en el piso, d) Enfrenta a los enemigos y, e) Golpea a la secretaria. Esto únicamente si el(la) jugador(a) identifica correctamente al culpable (solo si el(a) jugador(a) analiza correctamente tanto las conversaciones con los sospechosos como las pistas que le brindan los *stakeholders*: experta forense y periódicos) ya que en caso contrario únicamente se podrá llevar a cabo la primera misión.

Como primer paso dentro de esta escena, el(la) jugador(a) deberá entrevistar a la testigo en busca de información que sea relevante para el caso. De igual forma, deberá hablar con la experta forense. Es importante mencionar que este paso es independiente de si el(la) jugador(a) identificó correctamente o no al culpable (ver Sección 3.5.5).

El segundo paso se llevará a cabo sí y solo sí se identificó correctamente al culpable, por lo que el(la) jugador(a) deberá realizar cuatro diferentes misiones: la primera consiste únicamente en llegar a la estación de trenes, la segunda en prepararse para la batalla contra los enemigos -para esto, el(la) jugador(a) deberá de tomar una espada, la cual se va a encontrar clavada en el piso-, como tercera misión el(la) jugador(a) deberá enfrentar uno a uno a los villanos (los cuales lo perseguirán y atacarán) -en esta misión es importante tener cuidado de no morir, ya que, en caso de hacerlo, el juego se terminará y, por ende, el(la) jugador(a) perderá, por lo que no podrá llevar a cabo la última misión que consiste en golpear al culpable -sí y solo sí el(la) jugador(a) vence a todos los enemigos- ganando así el juego.

3.5.6.1. Diseño del escenario

Esta escena incluye los siguientes elementos:

- *Estación de trenes.* Contiene trenes, rieles, montañas, árboles, tráileres, faroles, un puente, casas, madera, vagones, suministros de carbón, torres de vigilancia, plumas y tanques de agua.

- *Área de batalla*. Contiene cuatro paredes transparentes las cuales fungen como limitadores del área de combate.

3.5.6.2. Personajes

Los personajes presentes dentro de esta escena son los siguientes: testigo, detective, secretaria, villano 1, villano 2, y villano 3.

3.5.7. C2_4_Cuarto_Cámaras

En esta escena, el(la) jugador(a) deberá realizar cuatro misiones. La primera consiste únicamente en llegar al cuarto de cámaras. La segunda misión consiste en comprobar las cámaras de seguridad, esto con el fin de darse cuenta de que los videos de las cámaras fueron robados. La tercera misión consiste en hablar con el fisgón, es importante prestar atención a esta conversación, ya que este le compartirá al(a) jugador(a) información importante que lo ayudará con el avance del caso. Es importante tener en cuenta también que, en caso de que el(la) jugador(a) no recuerde la conversación, podrá apoyarse en el botón “Resumen de diálogos”, el cual estará disponible durante todo el juego. Sin embargo, la información se va a desbloquear solo cuando el(la) jugador(a) hable con los *stakeholders* clave para la resolución del caso. Como última misión, una vez que el(la) jugador(a) haya conseguido los videos robados de las cámaras de seguridad, deberá ir al cuarto de cámaras y reproducir el video para ver quién es el culpable.

3.5.7.1. Diseño del escenario

Esta escena incluye los siguientes elementos: servidores, puerta, extintores, escritorio, monitores, cámaras de seguridad, compuertas de ventilación, silla, ratón de computadora, teclado, taza de café, porta lápices, lápices, CPUs, cajones, lector de huellas dactilares y letrero de salida.

3.5.7.2. Personajes

Los personajes presentes dentro de esta escena son los siguientes: fisgón y detective.

3.5.8. C2_5_Parque

En esta escena el(la) jugador(a) deberá realizar seis misiones. La primera consiste en interrogar al compañero de trabajo de la víctima, esto con la esperanza de encontrar información que ayude en la resolución del caso. La segunda y la cuarta misión consisten en comprar los periódicos del día del atentado y de un día anterior. La tercera y la quinta misión consisten en leer y analizar la información de cada uno de los dos periódicos. Por último, y después de haber analizado la información de todos los *stakeholders* identificados hasta ese momento, se deberá realizar la quinta misión, la cual consiste en darle al alcalde la lista de los tres posibles sospechosos.

3.5.8.1. Diseño del escenario

Esta escena incluye los siguientes elementos:

- *Quiosco de periódicos*. Mesas, periódicos, notas, cómics, revistas y letreros.
- *Parque*. Bancas para comer, bancas para sentarse, dispensador de refrescos, contenedor de basura, bolsa de basura, carro de hamburguesas, carrito del súper, flores, plantas y contenedores de agua.

3.5.8.2. Personajes

Los personajes presentes dentro de esta escena son los siguientes: NPC, compañero de trabajo de la víctima, y detective.

3.5.9. C2_6_Parque de juegos

En esta escena el(a) jugador(a) deberá llevar a cabo una sola misión: entrevistar al alcalde con el fin de obtener información que ayude en la resolución del caso.

3.5.9.1. Diseño del escenario

Esta escena incluye los siguientes elementos:

- *Parque de juegos*. Que contiene parcelas de cosechas de lechuga, tomate y calabaza, piedras, flores, árboles, vacas, borregos, patos y pollos.
- *Área de juegos*. Con árboles, mesas, columpios, resbaladillas, sube y baja, pasamanos, areneros, escaleras, flores, bancas, balones, piedras y macetas.

3.5.9.2. Personajes

Los personajes presentes dentro de esta escena son los siguientes: alcalde y detective.

3.5.10. C2_7_Límites_Ciudad

En esta escena se llevan a cabo dos misiones, la primera es ir a los límites de la ciudad y buscar al informante anónimo que llamó al(a) detective. Para poder llevar a buen término esta misión, es importante que el(la) jugador(a) tenga dinero suficiente, ya que el informante se lo pedirá a cambio de entregar los videos de las cámaras que fueron robadas. Como segunda misión, el(la) jugador(a) deberá comprar las cintas de video que le ofrece el informante anónimo. En caso de que se tenga el dinero suficiente, el(la) jugador(a) obtendrá un saldo negativo, lo cual le representará problemas en el futuro.

3.5.10.1. Diseño del escenario

Este escenario contiene edificios, casas y una parada de autobús.

3.5.10.2. Personajes

Los personajes presentes dentro de esta escena son los siguientes: informante anónimo, pollo, NPC 1, NPC 2, y detective.

3.5.11. C2_8_Área_Residencial

En esta escena se deberán realizar dos misiones, la primera consiste en llevarle la lista de sospechosos al alcalde. Para poder realizarla, el(la) jugador(a) deberá interactuar con él y con la secretaria, es importante que se preste atención en esta parte, porque la información que se revelará es de vital importancia para resolver el caso. Dentro de esta misión, el(la) jugador(a) tendrá la opción de cambiar su lista de sospechosos. En la segunda misión, recibirá una llamada sospechosa, la cual deberá contestar en un lugar alejado del alcalde y la secretaria.

3.5.11.1. Diseño del escenario

Dicho escenario incluye elementos como edificios habitacionales, edificios comerciales y casas.

3.5.11.2. Personajes

Los personajes presentes dentro de esta escena son los siguientes: secretaria, alcalde, vacas, ovejas, pollos, y detective.

3.5.12. C2_9_Final_Perdiste

Esta escena representa el primer final del juego y ocurrirá si el(la) jugador(a) no identifica correctamente al culpable después de haber reproducido la cinta de video, robada del cuarto de cámaras.

3.5.12.1. Diseño del escenario

El escenario del crimen incluye los siguientes elementos: cinta de video, un cadáver, un arma, un cono y balas en el piso.

3.5.12.2. Personajes

Los personajes presentes dentro de esta escena son la secretaria y el(la) detective.

3.5.13. C2_10_Final_Moriste

Esta escena representa el segundo final de la historia, cuando el(la) jugador(a) muere al dejar que los villanos lo derroten en el combate. Por lo que, perderá automáticamente la partida de juego.

3.5.13.1. Diseño del escenario

El escenario incluye los siguientes elementos: tumba, rieles, banqueta y trenes.

3.5.13.2. Personajes

Los personajes presentes dentro de esta escena son los siguientes: secretaria, villano 1, villano 2, villano 3, y detective.

3.5.14. C2_11_Final_Ganaste

Esta escena representa el tercer y último final posible en la historia y sucede cuando el(la) jugador(a) vence a los villanos y procede a golpear al culpable. Cuando se realice esta acción, el(la) jugador(a) automáticamente ganará el juego.

3.5.14.1. Diseño del escenario

Este escenario incluye elementos como árboles, rieles, vagones, madera, carretera y el cuarto de cámaras.

3.5.14.2. Personajes

Los personajes presentes dentro de esta escena son los siguientes: vecino conflictivo, testigo, fisgón, policía, experta forense, mejor amigo, NPC tutorial, y detective.

3.5.15. C2_12_Plantilla_De_Volere

Esta escena hace una representación, contextualizada a la historia presentada en el JS, de la plantilla de *Volere* (Robertson y Robertson, 2012). Dicha plantilla irá cambiando de acuerdo con el avance que se lleve en el juego y a los *stakeholders* que hayan sido descubiertos durante el desarrollo de las misiones.

3.5.15.1. Diseño del escenario

Este escenario incluye los siguientes elementos de interfaz gráfica:

- *Imagen que representa a la platilla de Volere.* Esta imagen es la representación de la plantilla de *Volere* (Robertson y Robertson, 2012) y contiene a todos los *stakeholders* clave (i.e., detective, policía, experta forense, alcalde, cintas de video, fisgón, testigo, periódico e informante anónimo) para resolver el caso.
- *Imágenes interrogativas.* Estas imágenes tienen la intención de ocultar la identidad de cada uno de los *stakeholders* involucrados en el caso, cada una de estas imágenes irá revelándose conforme se les vaya entrevistando.

3.6. Fase 3: Identificación/etiquetado de los retos educativos y evaluación

La Tabla 2 resume los retos educativos que el(la) jugador(a) aborda durante las sesiones de juego.

Tabla 2. Retos educativos y forma de evaluación dentro del juego serio

Elemento del juego	Reto/Desafío educativo
Personajes, documentos y videos	Identificar y consultar todas las posibles fuentes de requisitos (<i>stakeholders</i>)
Identificar correctamente al culpable	Identificar clases de usuarios y sus características
Diálogos e interrogatorios	Identificar y consultar con los actores del sistema
Plantilla de <i>Volere</i>	Utilizar esquemas para caracterizar y evaluar las relaciones apropiadas entre todos los <i>stakeholders</i>

Los retos establecidos son descritos en las siguientes secciones del documento.

3.6.1. Identificar y consultar todas las posibles fuentes de requisitos (*stakeholders*)

Este reto educativo consiste en que el(la) jugador(a) identifique quiénes son los *stakeholders* que le proporcionarán información relevante para la resolución del caso. De igual forma, se busca que lo(a)s estudiantes comprendan que existen diferentes tipos de *stakeholders* y que se hagan a la idea de que no solamente las personas pueden proporcionar información relevante sobre el producto de *software* que se desea desarrollar, sino que también los documentos u otros medios, como lo son los periódicos, notas y videos dentro del JS, pueden brindar información importante.

Para cumplir con este objetivo, el(a) jugador(a) deberá recorrer las diferentes escenas y entablar una conversación con los *stakeholders* que se encontrará en cada una de las misiones, también deberá analizar la información dentro de los periódicos, videos y notas con el fin de recoger información que lo lleve a atrapar al culpable.

3.6.2. Identificar clases de usuarios y sus características

Este reto educativo tiene como objetivo ayudar al(a) jugador(a) a distinguir quiénes pueden ser los *stakeholders* potenciales que empaten con las características del culpable. Dentro de este contexto, cada vez que el(la) jugador(a) entreviste a los *stakeholders*, los lea (en caso de periódicos y notas) o los vea (en caso de los videos) deberá prestar suma atención, esto con el fin de obtener información que ayude con la resolución del caso y de poder así, identificar a los *stakeholders* potenciales que puedan cumplir el rol de sospechosos.

3.6.3. Identificar y consultar con los actores del sistema

Este reto consiste en identificar a los *stakeholders*, entablar con ello(a)s una conversación (si procede), o en su caso leerlos o analizarlos, con el fin de obtener información que sea de ayuda para la resolución del caso. Para que este reto se lleve a cabo, es necesario estar a cierta distancia del *stakeholder* y se debe de pulsar en el teclado la tecla “Espacio”, para empezar una conversación con una persona, o para poder ver un video, en caso de las notas se pulsará esta misma tecla para agarrarlas, posteriormente se deberá acceder al inventario para leerlas.

3.6.4. Utilizar esquemas para caracterizar y evaluar las relaciones apropiadas entre todos los *stakeholders*

Para que este reto se cumpla, es necesario visualizar la plantilla de *Volere* (Robertson y Robertson, 2012) mediante un diagrama esquematizado mostrando el rol que va a desempeñar cada *stakeholder* dentro del juego y sus relaciones con los demás *stakeholders*. Para acceder a esta plantilla, el(la) jugador(a) cuenta con dos opciones: la primera es acceder manualmente a ella por medio de un botón que se mostrará en la interfaz gráfica del juego, y la segunda, donde se abrirá automáticamente mostrándose en pantalla cuando el(la) jugador(a) entable una conversación con un *stakeholder* que brinde información importante para la resolución del caso.

3.7. Fase 4: Identificación/clasificación de emociones

Almeida (2019) afirmó que la Inteligencia Emocional (IE) es fundamental para la formación, el desarrollo y el mantenimiento tanto de las relaciones personales como profesionales. Además, demostró, a través de la experimentación con un JS, que existe una relación entre las habilidades de IE del(a) jugador(a) y su desempeño dentro del juego. Por tal motivo, es posible afirmar que las emociones de lo(a)s jugadores son importantes para mejorar la jugabilidad del juego, de tal forma que mientras mayor impacto emocional exista dentro de cada escena, mayor será la motivación del(a) estudiante para seguir jugando. Dentro del contexto particular del juego presentado en esta Tesis, las emociones presentes dentro de cada una de las escenas y que pretenden causar un impacto dentro del personaje, son las siguientes:

3.7.1. Emociones presentes en la escena C1_1_Tutorial

- Intriga. Se busca que el(la) jugador(a) se muestre intrigado(a) por saber la temática del juego, su contenido, los controles, para qué sirve cada botón, etc.
- Curiosidad. Se busca que el(la) jugador(a) sienta interés por el entorno que lo(a) rodea de tal forma que se muestre curioso(a) por los objetos presentes dentro de cada una de las escenas y tenga la necesidad de interactuar con ellos.
- Indiferencia. Puede suceder que el(la) jugador(a) no se sienta sorprendido(a) ni emocionado(a) por jugar el juego.

3.7.2. Emociones presentes en la escena C1_2_Escoger_Avatar

- Duda. Dentro de esta escena se puede escoger un personaje de tres disponibles, por lo que es probable que el(la) estudiante se muestre dudoso(a) de su elección, ya que una vez que el avatar sea seleccionado, no será posible cambiarlo.

3.7.3. Emociones presentes en la escena C2_0_Entrada_Ciudad

- Miedo. Puede suceder que el(la) estudiante no haya prestado la suficiente atención al tutorial, por lo que existe la posibilidad de que sienta miedo de no poder recordar cómo es la jugabilidad correcta del juego.
- Entusiasmo. Se busca que el(la) estudiante se entusiasme por jugar.
- Alegría. Se busca que el escenario sea del agrado del(a) estudiante y disfrute de él.
- Curiosidad. El objetivo principal del juego es realizar las misiones y terminarlo, sin embargo, también se busca que el(la) jugador(a) se sienta curioso(a) por los objetos y los NPC que verá durante del juego.
- Fastidio. El juego cuenta con varios sonidos alrededor de las escenas (e.g., pasos, medio ambiente, autos), los cuales, al ser repetitivos, pueden llegar a molestar al(a) jugador(a), ocasionando que obtenga un bajo rendimiento durante la sesión de juego.

3.7.4. Emociones presentes en la escena C2_1_Casa_Víctima

- Impaciencia. Es importante mencionar que cada una de las escenas dentro del juego están separadas por una distancia geográfica significativa, y que las misiones no se llevan a cabo en orden. Esto significa que el(la) jugador(a) cambiará continuamente de escenas, lo que puede ocasionar que se sienta impaciente, ya que tendrá que recorrer varias veces el mismo camino sin poder aumentar su velocidad.
- Desesperación. Dentro de esta escena el(la) jugador(a) tiene que encontrar cuatro diferentes notas, por lo que puede desesperarse en caso de no encontrarlas rápidamente.
- Curiosidad. Se busca que el(la) jugador(a) se sienta curioso(a) por saber cuál es el contenido de las notas.

3.7.5. Emociones presentes en la escena C2_2_Morgue

- Curiosidad. Puede suceder que el(la) jugador(a) sienta curiosidad al hablar con la experta forense, ya que es ella quien le va a proporcionar los primeros datos relevantes para la resolución del caso.
- Sorpresa. Puede ser que el(la) jugador(a) sienta sorpresa al estar dentro la morgue, ya que es aquí donde se localiza el cadáver de la víctima y donde podrá observar la escena del crimen.

3.7.6. Emociones presentes en la escena C2_3_Estación_Trenes

- Estrés. Al tener que pelear con tres diferentes enemigos, el(la) jugador(a) puede sentirse estresado(a) ya que estos personajes lo van a perseguir porque quieren matarlo(a).
- Felicidad. Puede suceder que el(la) jugador(a) sienta felicidad al hablar con la testigo, ya que ésta le va a proporcionar datos relevantes para la resolución del caso.
- Tristeza. Es posible que el(la) jugador(a) sienta tristeza, ya que los villanos lo(a) puedan matar.
- Alegría. Puede que el(la) jugador(a) sienta alegría, ya que una vez que acabe con los enemigos y encuentre al culpable, ganará el juego.

3.7.7. Emociones presentes en la escena C2_4_Cuarto_Cámaras

- Sorpresa. Una de las misiones que se realizan en esta escena consiste en revisar las cámaras de seguridad; sin embargo, alguien las robó. Por lo que, al poner la cinta, se mostrará la pantalla con diferentes colores y ruido y, si el(la) jugador(a) se queda observando por más de cinco segundos, podrá ver un video animado que tiene como propósito hacer que se sienta sorprendido. También, se busca que el(la) jugador(a) se sorprenda al enterarse de quien es el culpable.
- Entusiasmo. Se busca que el(la) jugador(a), en caso de que se quede viendo el video de las cámaras por más de cinco segundos, se entusiasme al ver la evidencia que en él se encuentra.

3.7.8. Emociones presentes en la escena C2_5_Parque

- Temor. Puede que el(la) jugador(a) sienta temor porque en esta escena tiene que escribir la lista con los tres principales sospechosos y, en caso de escoger mal al culpable, podría desencadenar graves consecuencias dentro del juego.
- Entusiasmo. El(a) jugador(a) puede sentirse entusiasmado(a) al leer los periódicos, ya que estos le van a proporcionar datos relevantes para la resolución del caso.

3.7.9. Emociones presentes en la escena C2_6_Parque de juegos

- Intriga. El(la) jugador(a) puede sentirse intrigado(a) con esta escena ya que contiene varios elementos visuales atractivos (e.g., columpios, pelotas, pasa manos, resbaladillas, bancas, areneros, sube y baja) que lo(a) pueden incentivar a explorarla.

3.7.10. Emociones presentes en la escena C2_7_Límites_Ciudad

- Molestia. Es posible que el(la) jugador(a) se sienta molesto(a) al tener que comprar a un alto precio los videos robados del edificio de cámaras.
- Felicidad. El(la) jugador(a) podrá sentirse feliz al descubrir quién es el culpable.

3.7.11. Emociones presentes en la escena C2_8_Área_Residencial

- Felicidad. Puede suceder que el(la) jugador(a) sienta felicidad al hablar con el alcalde y la secretaria, ya que estos le proporcionarán pistas para dar con la identidad del posible culpable.

3.7.12. Emociones presentes en la escena C2_9_Final_Perdiste

- Frustración. Puede suceder que el(la) jugador(a) se sienta frustrado(a), ya que no pudo identificar correctamente al *stakeholder*.
- Enojo. El(la) jugador(a) podrá sentir enojo por no haber tenido un mejor desempeño dentro del juego, además la culpable se estará burlando ya que no pudo atraparla.
- Tristeza. El(la) jugador(a) podrá experimentar tristeza, ya que no pudo seguir avanzando en el juego debido a que este se terminó por una mala identificación del culpable.
- Vergüenza. Puede que el(la) jugador(a) tenga este sentimiento al ver como lo(a)s demás jugadores han avanzado a la siguiente fase del juego y él no, debido a que no identificó correctamente al culpable.

- Desinterés. Puede que el(la) jugador(a) no haya sentido una conexión ni atracción por el juego, por lo que perder no le será relevante.

3.7.13. Emociones presentes en la escena C2_10_Final_Moriste

- Frustración. El(la) jugador(a) puede sentir frustración al saber que todo el avance que tuvo a lo largo del juego fue en vano, ya que al final perdió el combate y, por ende, el juego.
- Enojo. Puede que el(la) jugador(a) sienta enojo al saber que fue derrotado(a) por los villanos y reciba, como consecuencia, burlas de parte de éstos.
- Tristeza. Es probable que el(la) jugador(a) se sienta triste debido a que se esforzó durante todo el juego y al final no ganó.
- Desinterés. Puede que el(la) jugador(a) no haya mostrado interés en el juego, por lo que una victoria o una derrota no le será relevante.

3.7.14. Emociones presentes en la escena C2_11_Final_Ganaste

- Alegría. Puede que el(la) jugador(a) sienta alegría al saber que ganó el juego.
- Desinterés. Puede que el(la) jugador(a) no haya sentido emoción alguna al interactuar con el juego, por lo que una victoria dentro del juego no le resultaría importe.

3.7.15. Emociones presentes en la escena C2_12_Plantilla_De_Volere

- Confusión. Puede que el(la) jugador(a) sienta confusión la primera vez que vea la plantilla de *Volere*, y no le entienda.
- Felicidad. Puede que el(la) jugador(a) se muestre feliz al saber que está comprendiendo de mejor manera un tema previamente visto en clase.

3.8. Fase 5: Diseño de la adaptación

El(la) jugador(a) tendrá la libertad de seleccionar su avatar, además podrá personalizarlo por medio de la tienda incluida en el juego, ya que podrá adquirir accesorios como “mascotas” y sombreros. Sin embargo, deberá tener cuidado de no comprar de más, ya que, en cierto momento, el(la) jugador(a) puede tener saldo negativo, lo que repercutirá en el desarrollo del juego puesto que no tendrá el dinero suficiente para realizar algunas misiones.

Por otra parte, el desarrollo del juego fue adaptado a una resolución de pantalla con dimensiones de 1280 x 720, mismas que pueden ser cambiadas. Sin embargo, esto podría afectar la apariencia visual del juego, así como algunas de sus funcionalidades, por lo que no se recomienda utilizar otras resoluciones.

3.9. Fase 6: Diseño de la colaboración

Este JS fue diseñado para jugarse de manera individual, por tal motivo, esta fase de la metodología no fue implementada.

3.10. Implementación del videojuego

La implementación de este juego fue realizada en 3D, lo que significa que tanto los personajes como los objetos utilizados dentro de éste cuentan con tres dimensiones: el eje x , y y z . Además, se consideró un diseño en tercera persona con el fin de que el(la) jugador(a) pudiera apreciar su avatar y los diferentes accesorios que podrá comprar en la tienda. A continuación, se detalla el proceso de desarrollo del juego considerando la metodología presentada en la Figura 3.1.

3.10.1. Herramientas y lenguaje de programación

Este juego se implementó utilizando la herramienta Unity® mediante un proyecto 3D. Se seleccionó esta herramienta debido a que ha demostrado ser adecuada para la implementación de juegos en 3D. Además, cuenta con su propia tienda de *assets*¹³, lo que facilita la elaboración de los juegos al proveer objetos y entornos. En cuanto al lenguaje de programación se utilizó C#, ya que es uno de los lenguajes más populares con los que trabaja Unity®.

3.10.2. Desarrollo

Para la elaboración del juego fue necesaria la identificación de los retos educativos y su relación con las actividades dentro del mismo. Una vez definidos, se continuó con el diseño de los dos capítulos del juego y sus respectivas escenas.

Como primer paso se implementó el menú de inicio del juego (ver Figura 3.11). El cual puede verse como el comienzo de éste, ya que es donde el(la) jugador(a) podrá iniciar una partida dando *clic* al botón de “Inicio”, acceder a la información sobre los *assets* utilizados dentro del juego dando *clic* en el botón de “Acerca de”, o bien salir del juego si así lo desea dando *clic* en el botón de “Salir”.

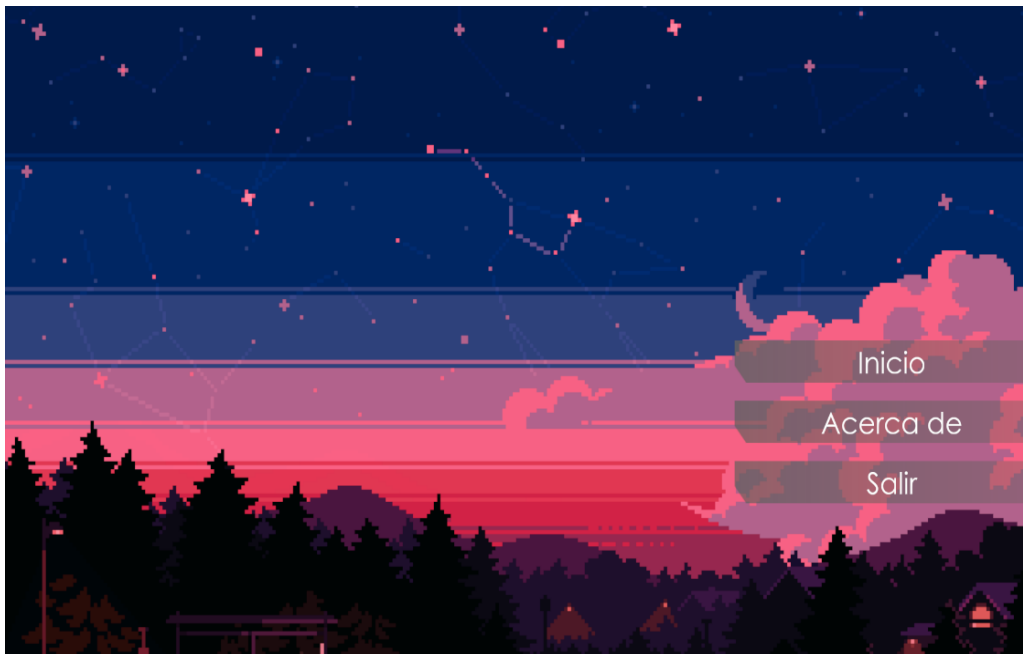


Figura 3.11. Pantalla de inicio del juego serio

¹³ Un *asset* es una representación de cualquier ítem que puede ser utilizado en un juego o proyecto (Unity, 2016).

Una vez que el(la) jugador(a) haga *clic* en el botón de “Inicio” deberá esperar unos segundos para empezar a jugar, debido a la carga interna de la computadora. Para tratar de paliar este problema se implementó una pantalla de carga a través de una animación que le indica al(a) jugador(a) que tendrá que esperar, haciendo así más amena la espera (ver Figura 3.12).

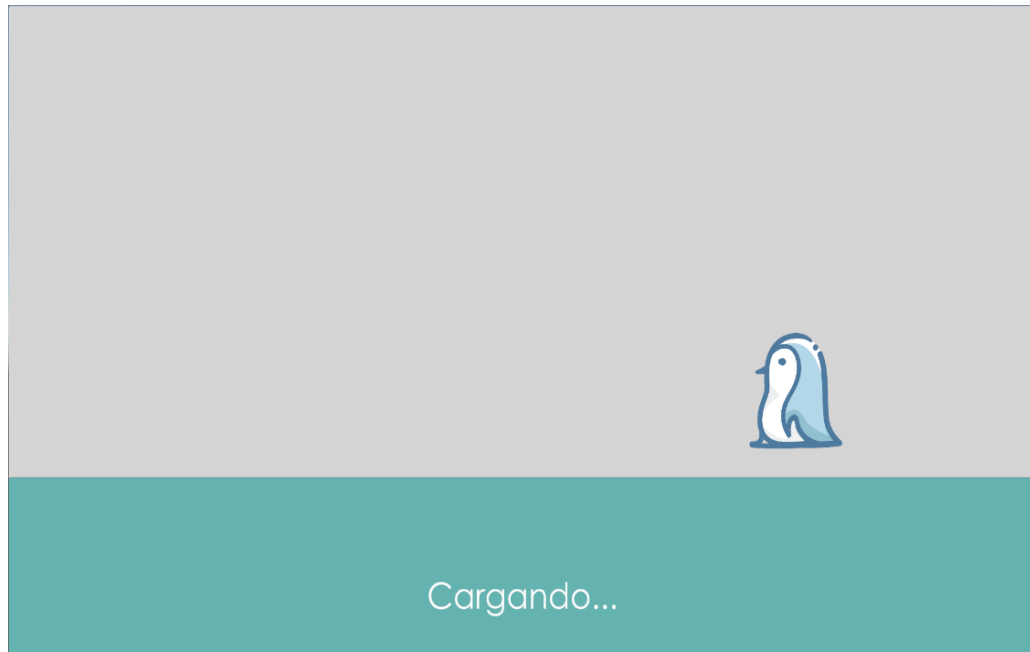


Figura 3.12. Pantalla de carga animada

Posteriormente, se iniciará la escena del tutorial que se muestra en la Figura 3.13.



Figura 3.13. Ejemplo del tutorial del juego

Esta escena tiene como objetivo proporcionar la información necesaria al(a) jugador(a) para que se comprenda la temática del juego, así como iniciarlo(a) en el uso de los controles. Una vez que termine de ver e interactuar con el entorno del tutorial, se deberá pulsar el *clic* izquierdo del ratón de la computadora para que se avance a la escena C1_2 _Escoger_Avatar (ver Figura 3.14).



Figura 3.14. Avatares disponibles en el juego

En este sentido, el(la) jugador(a) puede escoger entre tres diferentes personajes: un hombre, una mujer y un robot. Esta acción se realiza con el fin de que el(la) jugador(a) se sienta identificado(a) con el personaje mientras interactúa con el entorno del juego. Para hacer esta escena más interactiva, cada vez que el(la) jugador(a) seleccione un avatar, este se iluminará y comenzará a bailar. En caso de que el(la) jugador(a) no seleccione ningún avatar y de *clic* directamente en el botón de “Jugar”, se le asignará por default el personaje no binario. Una vez que el avatar ha sido escogido, el juego dará comienzo en la escena C2_0_Entrada_Ciudad (ver Figura 3.15).

Cada uno de los diálogos establecidos en el juego se presentan dentro de un panel, mismo que aparecerá cada vez que se empiece una conversación o cuando se dé una alerta durante el desarrollo de una misión (ver Figura 3.16). En este sentido y para hacer más interactivo el juego, cada vez que se empiece un diálogo, los personajes realizarán movimientos y emitirán sonidos con el fin de simular una conversación real.

Una vez que se termine con la misión asignada, el(la) jugador(a) deberá trasladarse a la siguiente escena. Al tratarse de un escenario lo suficientemente grande (i.e., ciudad de *Wonderland* y cercanías), fue necesario implementar un mapa que ayude al(a) jugador(a) a llegar a su destino. Siguiendo esta lógica, se implementó un mapa en la esquina superior izquierda de la pantalla del juego y, cada vez que se asigna una misión al(a) jugador(a), se habilita una línea roja que servirá como guía para que el(la) jugador(a) llegue a la ubicación en donde tiene que llevar a cabo la misión.

Por consiguiente, al ser un juego con muchos elementos visuales atractivos, existe la posibilidad de que el(la) jugador(a) se distraiga y se pierda, por tal motivo, tendrá siempre la opción de aumentar el tamaño del mapa haciendo *clic* sobre él. De esta manera, se podrá ubicar de mejor forma en la ciudad y volver a la línea que lo(a) lleve a su destino (ver Figura 3.17).



Figura 3.15. Inicio del juego en la ciudad de *Wonderland*



Figura 3.16. Ejemplo de diálogos en entrevistas y/o interrogatorios

Es importante mencionar que, para que cada misión se cumpla correctamente, es necesario que el(la) jugador(a) llegue al punto que especifica la línea roja. Por lo que, para facilitar el cumplimiento de dichas misiones, cada uno de los puntos específicos a los que se debe llegar se resaltan con color amarillo y una flecha roja que tiene un movimiento constante.



Figura 3.17. Ampliación del mapa de la ciudad de *Wonderland*

Ahora bien, cuando el(la) jugador(a) llegue a la siguiente escena (C2_1_Casa_Víctima), deberá realizar las misiones correspondientes que consisten en entablar una conversación con la esposa de la víctima, buscar y, posteriormente, leer las notas de la agenda (ver Figura 3.18).



Figura 3.18. Búsqueda de información en la casa de la víctima

Para implementar la dinámica de la búsqueda de notas, se colocó una variedad de documentos en diferentes partes de la casa, que no necesariamente son útiles, por lo que el(la) jugador(a) deberá encontrar las correctas y guardarlas dentro del inventario. Cada que el(la) jugador(a) encuentre una nota, deberá acercarse lo suficiente hasta que ésta cambie de color, lo cual va a indicarle que la puede agarrar (ver Figura 3.19). Ahora bien, para agarrar y guardar la nota en el inventario, el(la) jugador(a)

deberá pulsar la tecla “Espacio”. Posteriormente, podrá acceder al inventario pulsando la tecla I (tanto para abrirlo como para cerrarlo) (ver Figura 3.20). Esta opción se encuentra presente de manera permanente en la parte inferior derecha de la interfaz gráfica, con lo que se busca que el(la) jugador(a) no olvide esta acción, ya que es necesario leer las notas para que la jugabilidad fluya correctamente.



Figura 3.19. Identificación de documentos en el juego



Figura 3.20. Inventario de objetos recogidos en el juego

Una vez terminada la misión, se deberá ir a las escenas C2_2_Morgue, C2_6_Parque de juegos, C2_5_Parque y C2_1_Casa_Víctima con el fin de entrevistar a los personajes implicados en las pesquisas. Posteriormente, el(la) jugador(a) deberá dirigirse a la escena C2_4_Cuarto_Cámaras, en donde reproducirá el video de las cámaras y hablará con el fisgón (ver Figura 3.21).



Figura 3.21. Búsqueda de pistas en el cuarto de cámaras

Para continuar, el(la) jugador(a) deberá dirigirse a la escena C2_3_Estación_Trenes con el fin de interrogar a la testigo (ver Figura 3.22).



Figura 3.22. Estación de trenes

Con la información que le proporcione, el(la) jugador(a) deberá dirigir a la escena C2_5_Parque en busca de los periódicos de uno y cinco días antes del asesinato para comprarlos. Estos documentos deberán analizarse para obtener pistas del posible culpable. La misión concluye cuando el(la) jugador(a) seleccione a tres sospechosos de su lista (ver Figura 3.23).



Figura 3.23. Lista de sospechosos

Al tener la lista de posibles culpables, el(la) jugador(a) deberá trasladarse a la escena C2_8_Área_Residencial, con el fin de entregársela al alcalde. Sin embargo, al estar interactuando con este personaje, su secretaria llega a darle información que es crucial para resolver el caso, por lo que el(a) jugador(a) deberá estar atento(a). En este sentido, se tiene la oportunidad de cambiar la lista o entregarla como ya se tenía. Una vez terminada esta misión, el(la) jugador(a) recibirá una llamada de un informante anónimo que le indicará que se tiene que dirigir en la ubicación que corresponde a la escena C2_7_Límites_Ciudad. Este personaje ofrecerá a la venta un video inédito de la cámara de seguridad que demuestra quien es el culpable. Sin embargo, en caso de que el(la) jugador(a) haya escogido incorrectamente al culpable de su lista de sospechosos, el juego llegará a su fin y el caso no será resuelto.

Por otro lado, en caso de que el(la) jugador(a) haya identificado correctamente al culpable, deberá dirigirse a la ubicación de la escena C2_3_Estación_Trenes para enfrentarse al asesino y a sus secuaces. Durante el enfrentamiento, los villanos podrán atacar uno a uno al(a) jugador(a) por un tiempo determinado. Cada uno de los villanos tiene diferentes armas y movimientos, siendo uno más difícil que el otro. El(la) jugador(a) cuenta con un total de 100 puntos de vida que se le asignan al inicio de la pelea y perderá dos por cada golpe que reciba. Mientras que los villanos perderán un punto de vida por cada golpe recibido. Esto con el fin de que lo(a)s estudiantes que no estén familiarizados con los juegos, no pierdan fácilmente. Es importante enfatizar que esta parte del juego conduce a dos diferentes finales representados por (1) la escena C2_10_Final_Moriste, en donde los villanos bailan al lado de la tumba en caso de que el(la) jugador(a) pierdan la pelea, y (2) la escena C2_11_Final_Ganaste, que consiste en que los pobladores de la ciudad de *Wonderland* celebran la victoria del(a) detective al derrotar a los villanos y resolver el crimen.

3.10.3. Elementos de apoyo extras dentro del videojuego

En caso de que el(la) jugador(a) no recuerde información al jugar, podrá usar el botón de “Resumen de diálogos” para recibir información relevante de algunos *stakeholders* (ver Figura 3.24), o bien el botón “Misión actual” para recibir información al respecto (ver Figura 2.35).



Figura 3.24. Resumen de diálogos



Figura 3.25. Resumen de misión actual

El(la) jugador(a) tiene la opción de personalizar su avatar con accesorios y mascotas que podrá adquirir en la tienda con el dinero acumulado por terminar correctamente las misiones. Esto hará que el(la) estudiante se sienta más cómodo(a) e identificado(a) con el personaje (ver Figura 3.26).



Figura 3.26. Elementos disponibles en la tienda

Finalmente, el juego cuenta con un menú de pausa que es útil cuando el(la) jugador(a) desea detener la interacción con el juego (ver Figura 3.27).



Figura 3.27. Menú de pausa

4. Evaluación empírica

Como se mencionó anteriormente, para probar la efectividad del juego se realizó una evaluación empírica en un entorno educativo real que involucró el desarrollo de un caso de estudio con estudiantes de la Universidad Tecnológica de la Mixteca. A continuación se presentan los detalles y resultados obtenidos con dicha evaluación.

4.1. Estrategia de evaluación

La evaluación se llevó a cabo con 19 estudiantes de décimo semestre (generación 2019-2023) de la carrera de Ingeniería en Computación de la Universidad Tecnológica de la Mixteca y 5 egresados más de la misma carrera (generación 2018-2022). El total de lo(a)s estudiantes fue dividido en dos grupos: uno de control y otro experimental. Todo(a)s los estudiantes recibieron una clase teórica que incluyó tópicos como “Contexto e importancia de la IR”, “El proceso de elicitación de requisitos”, y la fase de “Identificación de los *stakeholders*”. Una vez que la clase concluyó, lo(a)s estudiantes del grupo de control respondieron un cuestionario para evaluar la comprensión de los temas y, por ende, obtener una medida de desempeño académico. La dinámica con los estudiantes del grupo experimental cambió, puesto que, antes de responder el cuestionario de desempeño académico, realizaron una sesión práctica con el juego “*Misterio en Wonderland*” para reafirmar el entendimiento de los conceptos expuestos durante la clase teórica. Finalmente, los datos recogidos durante la evaluación fueron analizados mediante estadística descriptiva para esbozar conclusiones que permitieran aceptar o rechazar la hipótesis establecida al inicio de esta Tesis.

4.2. Participantes

Como se mencionó anteriormente, 24 estudiantes participaron en la evaluación empírica del JS. Tanto el grupo de control como el experimental estuvieron integrados por 12 estudiantes que fueron escogidos al azar. Es importante mencionar que todo(a)s contaban con conocimientos previos sobre Ingeniería de *Software* e Ingeniería de Requisitos, puesto que ya habían tomado ambos cursos. Además, la estudiante que presenta esta Tesis asesoró al grupo experimental durante la sesión de juego.

Por otro lado, la directora de esta Tesis fue la responsable de impartir la clase teórica a lo(a)s estudiantes de ambos grupos, haciendo énfasis en la identificación de los *stakeholders* en el contexto del desarrollo de proyectos de *software*. Esta participante tiene una experiencia de más de 20 años haciendo investigación sobre la IS, específicamente sobre la IR.

Finalmente, un investigador con una experiencia de más de 15 años en la enseñanza de la Ingeniería de *Software* participó en la evaluación empírica revisando las respuestas de lo(a)s estudiantes y las conclusiones que se obtuvieron a partir de éstas.

4.3. Instrumentos

Con el objetivo de evaluar la efectividad del JS, en términos de aprendizaje, se estableció que ambos grupos de estudiantes (i.e., control y experimental) realizaran un examen de conocimientos posterior a la clase teórica para obtener información sobre su aprendizaje relacionado con la identificación correcta de los *stakeholders*. En este sentido, se diseñó un cuestionario de 15 ítems que se enfoca en temas específicos para la enseñanza de la IR a nivel licenciatura sugeridos en los Lineamientos Curriculares para Programas de Licenciatura en Ingeniería de *Software* (SE, 2014), en el Modelo de Competencias de la Ingeniería de *Software* (SWECOM, 2014) y en el Plan de Estudios de Computación 2020 (ACM-CC, 2020), los cuales, además, son expuestos durante la clase teórica (ver Tabla 3).

Tabla 3. Cuestionario diseñado para evaluar los conocimientos de los estudiantes

Pregunta	Respuesta
1. ¿Qué conocimiento deben tener los ingenieros de <i>software</i> para llevar a cabo de manera correcta la obtención de los requisitos?	
2. ¿Cuál es la primera medida de éxito en el desarrollo de <i>software</i> ?	
3. ¿Quiénes son las fuentes de requisitos?	
4. Proporciona ejemplos de posibles fuentes de requisitos.	
5. En el contexto de la Ingeniería de Requisitos, ¿a quiénes se les conoce como “ <i>stakeholders</i> ”?	
6. ¿Quiénes son los <i>stakeholders</i> que siempre están presentes en el desarrollo de <i>software</i> ?	
7. Los <i>stakeholders</i> participan únicamente al inicio de un proyecto, una vez terminada la fase de requisitos, esto(a)s no vuelven a involucrarse. Responde FALSO o VERDADERO y argumenta tu respuesta.	
8. ¿Es necesario que las partes interesadas cumplan un rol funcional dentro de un proyecto para ser consideradas como <i>stakeholders</i> ? Argumenta tu respuesta con un ejemplo.	
9. Los <i>stakeholders</i> con rol de clientes son los únicos que se deberán de asegurar que el producto cumpla con los estándares de calidad, ya que son ello(a)s los que están solicitando dicho producto. Responde FALSO o VERDADERO y argumenta tu respuesta.	
10. Los <i>stakeholders</i> negativos no son necesarios para el desarrollo de un proyecto, por lo que su participación puede ser descartada. Responde FALSO o VERDADERO y argumenta tu respuesta.	
11. La elección de los <i>stakeholders</i> depende del proyecto que se desea realizar. Responde FALSO o VERDADERO y argumenta tu respuesta.	
12. Existen guías o propuestas específicas que ayudan a identificar correctamente a los <i>stakeholders</i> dependiendo del tipo del proyecto. Responde FALSO o VERDADERO y argumenta tu respuesta.	

Pregunta	Respuesta
13. Los periódicos, notas y revistas pueden ser considerados como <i>stakeholders</i> . Responde FALSO o VERDADERO y argumenta tu respuesta.	
14. ¿Las personas afectadas por un desarrollo de <i>software</i> pueden considerarse como <i>stakeholders</i> ? Responde FALSO o VERDADERO y argumenta tu respuesta.	
15. ¿Se debe tomar en cuenta la motivación y área de experiencia de los <i>stakeholders</i> al identificarlos? Responde FALSO o VERDADERO y argumenta tu respuesta.	

Aunado a lo anterior, se adaptó el instrumento de medición de MEEGA+ (Petri et al., 2021) para evaluar la efectividad de los juegos digitales y no digitales considerando la usabilidad, confianza, desafío, satisfacción, interacción social, diversión, atención enfocada, relevancia, y aprendizaje percibido (ver Tabla 4). Es importante mencionar que “*Misterio en Wonderland*” no fue creado como un juego multijugador, por lo cual la dimensión de interacción social no fue incluida en la evaluación.

Tabla 4. Instrumento de medición de MEEGA+ (adaptado de Petri et al., (2021))

Dimensión por evaluar		Ítem	TD	ED	N	DA	TA	M	S
Usabilidad	Estética	1. El diseño del juego me pareció atractivo, respecto a la interfaz, gráficos, etc.							
		2. Las fuentes de texto y los colores me parecieron adecuados y consistentes.							
		3. Necesité usar el tutorial para comprender la temática del juego.							
	Capacidad de aprendizaje	4. Interpreté de forma fácil la jugabilidad.							
		5. Considero que la mayoría de personas puede aprender los comandos del juego de manera rápida.							
	Operabilidad	6. Pude jugar de forma fácil.							
		7. Las reglas del juego me parecieron claras y concisas.							
	Accesibilidad	8. El tamaño y estilo de las fuentes me fueron fáciles de leer.							
		9. Considero que colores fueron adecuados para la temática del juego.							
		10. Me fue posible personalizar la apariencia del juego (fuentes, colores, etc.) según mis preferencias.							
	Protección al usuario de posibles errores	11. El juego me impidió cometer errores.							
		12. Cuando cometí un error, me fue fácil recuperarme.							
Confianza	13. Desde el inicio del juego me dio la impresión de que iba a ser fácil de jugar.								

Dimensión por evaluar	Ítem	TD	ED	N	DA	TA	M	S
	14. El contenido y la estructura del juego me dieron la confianza de que aprendería algo nuevo.							
Desafío	15. Considero que la dificultad del juego fue la apropiada.							
	16. Considero que el ritmo de las misiones (situaciones, variaciones, nuevos obstáculos, etc.) iba variando adecuadamente conforme el tiempo avanzaba.							
	17. Considero que el juego y las misiones no se vuelven monótonas o aburridas a medida que avanza la historia.							
Satisfacción	18. El completar cada una de las misiones del juego me daba una sensación de logro.							
	19. Gracias a mi esfuerzo pude sortear los obstáculos del juego e ir avanzando.							
	20. Me siento satisfecho(a) con el aprendizaje obtenido.							
	21. Recomendaría este juego a mis amigo(a)s o compañero(a)s.							
Diversión	22. Me divertí con el juego.							
	23. Algunos elementos del juego (diálogos, misiones, videos, etc.) me hicieron reír.							
Atención enfocada	24. Hubo algo que llamó mi atención desde el inicio del juego.							
	25. Estaba tan concentrado(a) en las misiones del juego que perdí la noción del tiempo.							
	26. Me olvidé de mi entorno mientras jugaba.							
Relevancia	27. Considero que los contenidos del juego fueron relevantes.							
	28. Comprendí la relación del juego con los objetivos de aprendizaje.							
	29. El juego es una forma adecuada de proporcionar aprendizaje sobre el tema.							
	30. Prefiero aprender con este juego que con otras formas (libros, diapositivas, etc.).							
Aprendizaje percibido	31. El juego contribuyó a mejorar el aprendizaje del tópico visto en clase.							
	32. La enseñanza brindada por el juego fue más eficiente para el aprendizaje del tópico visto en clase que otros métodos tradicionales.							

TD= Totalmente en desacuerdo, ED = En desacuerdo, N= Neutral, DA= De acuerdo, TA= Totalmente de acuerdo, M= Media aritmética, SD= Desviación estándar.

Dicho instrumento contiene originalmente 35 ítems; sin embargo, con la adaptación realizada se utilizaron 32 finalmente. El total de los ítems se evalúa considerando una escala tipo *Likert* de cinco opciones: “Totalmente en desacuerdo” (**TD**)= -2, “En desacuerdo” (**ED**)=-1, “Neutral” (**N**)= 0, “De acuerdo” (**DA**)= 1 y “Totalmente de acuerdo” (**TA**)= 2, para su análisis posterior. Esto significa que las respuestas subjetivas se transforman en valores cuantitativos que permiten un análisis descriptivo mediante el cálculo de la media (**M**) y la desviación estándar (**SD**).

4.4. Desarrollo del caso de estudio

Como se indicó en el punto 4.2, la muestra del caso de estudio consistió en 24 estudiantes que conformaron al grupo de control (12 estudiantes) y al experimental (12 estudiantes). Ambos grupos recibieron una clase teórica siguiendo un enfoque tradicional con el objetivo de que lo(a)s estudiantes se relacionaran con conceptos básicos sobre la identificación correcta de los *stakeholders* en el contexto del desarrollo de proyectos de *software* (ver Figuras 4.1).



Figura 4.1. Estudiantes tomando la clase teórica como parte de la evaluación

Dicha clase se realizó en los Laboratorios de Posgrado de la Universidad Tecnológica de la Mixteca y tuvo una duración aproximada de 45 minutos. Durante este tiempo se pudo observar que, hasta los primeros 20 a 25 minutos, los estudiantes estuvieron atentos a la explicación de los temas y se mostraban cooperativos al momento de responder las preguntas que se realizaron en puntos cruciales de la exposición. Sin embargo, pasado este tiempo su atención comenzó a menguar y empezaron a distraerse dando paso al aburrimiento. De hecho, tres estudiantes mostraron poco interés en la clase desde su comienzo y prefirieron jugar con cosas que estaban a su alrededor. Algo que usualmente ocurre en una clase real.

Los tópicos impartidos durante la clase se agruparon en dos grandes temas, como se muestra a continuación:

1. Tema 1: Importancia de la Ingeniería de Requisitos.
 - i. ¿Qué son los requisitos?

- ii. Medidas de éxito de un proyecto de *software*.
 - iii. Causas de fracaso de los proyectos de *software* debido a requisitos mal elicitados.
 - iv. Importancia de la calidad en los requisitos de *software*.
2. Tema 2: Los *stakeholders* en la Ingeniería de Requisitos.
- i. ¿Qué son los *stakeholders*?
 - ii. ¿Quiénes pueden ser los *stakeholders*?
 - iii. Posibles fuentes de requisitos.
 - iv. Roles de un *stakeholder* dentro de un proyecto de *software*.
 - v. Conocimientos que debe poseer un *stakeholder*.
 - vi. *Stakeholders* más comunes en el desarrollo de *software*.
 - vii. Guías de apoyo para la identificación de *stakeholders*.
 - viii. Técnicas y métodos para identificar a los *stakeholders*.

A continuación se presentan los resultados obtenidos por ambos grupos de estudiantes, después de la impartición de la clase teórica.

4.4.1. Resultados obtenidos por el grupo de control

Al finalizar la clase, los participantes del grupo de control respondieron el cuestionario de evaluación para determinar su comprensión sobre los temas expuestos en la clase (ver Tabla 3). El análisis de sus respuestas condujo a la obtención de la gráfica mostrada en la Figura 4.2.

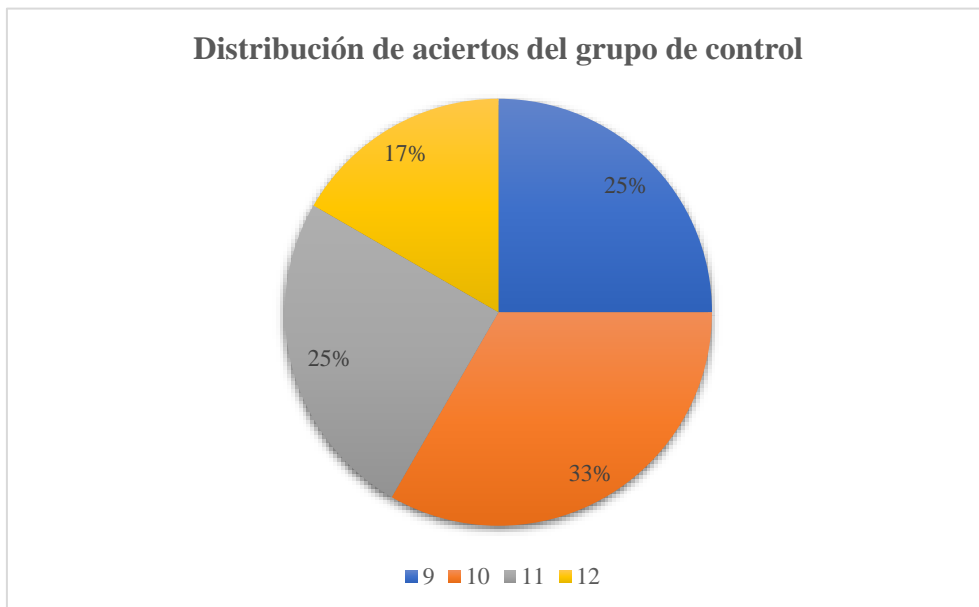


Figura 4.2. Evaluación del desempeño académico correspondiente al grupo de control

Como se puede observar en la Figura 4.2, el 25% de lo(a)s estudiantes en el grupo de control tuvo una puntuación de 9 aciertos correctos de 15, el 33% obtuvo 10 aciertos correctos de 15, el 25% acertó en 11 aciertos de 15 y, por último, el 17% logró obtener 12 aciertos de 15. Por lo tanto, el valor

medio fue de 10 aciertos correctos de 15, dando así una calificación promedio de 6.7/10. Es importante enfatizar que ningún estudiante del grupo de control logró los 15 aciertos.

4.4.2. Resultados obtenidos por el grupo experimental

Para llevar a cabo la evaluación del grupo experimental, los 12 estudiantes que conformaron este grupo permanecieron en los Laboratorios de Posgrado de la Universidad Tecnológica de la Mixteca, específicamente en el laboratorio de *Cómputo Inteligente e Inteligencia Artificial*, para iniciar la sesión de juego con *“Misterio en Wonderland”*. En este sentido, lo(a)s estudiantes seleccionaron aleatoriamente una computadora, a la que tuvieron que colocarle audífonos (los cuales les fueron solicitados con antelación) para crear una experiencia inmersiva con el juego. Cabe mencionar que el JS ya había sido instalado previamente en las computadoras y que se indicó a los estudiantes que la sesión de juego tendría una duración máxima de 60 minutos.

Como primer paso en el uso y posterior evaluación del juego, lo(a)s estudiantes tuvieron que hacer doble *clic* sobre el ícono del juego para visualizar el menú de inicio del mismo (ver Figura 3.11). Es importante mencionar que lo(a)s estudiantes se mostraron curioso(a)s por jugar argumentando que la estética del juego les parecía agradable. Posteriormente, se procedió a realizar las actividades del tutorial con el objetivo de relacionarse con los controles y el entorno del juego (ver Figuras 4.3). Las actividades del tutorial lograron llamar la atención de la mayoría de lo(a)s estudiantes; sin embargo, se pudo notar que alguno(a)s no prestaron la debida atención a pesar de que se les había indicado que la finalidad era familiarizarse con el juego para, posteriormente, iniciar la sesión de juego. En este sentido, esto(a)s estudiantes se distrajerón con los objetos y actividades de la escena a tal grado que uno(a) de ello(a)s hizo *clic* sobre algunos botones y comandos incorrectos que lo hicieron perderse por completo en la actividad. Afortunadamente, por el soporte que brindó la tesista durante la sesión de juego, se pudo recuperar la partida.

Una vez que lo(a)s estudiantes iniciaron formalmente la historia del juego, se hizo notoria la división entre lo(a)s que tenían experiencia previa con los videojuegos y lo(a)s que no. Esto es importante de resaltar, puesto que aquello(a)s que estaban acostumbrados a jugar no tuvieron problemas para interactuar con el juego, incluso pudieron terminar la sesión antes de lo esperado (~45 minutos). A diferencia de lo(a)s que no suelen jugar en la computadora, quienes terminaron la partida en un tiempo de 70 minutos. En consecuencia, esto(a)s estudiantes mostraron un poco de molestia y frustración en cuanto al uso de los controles y la interacción con el juego, por lo que realizaron más preguntas durante la sesión de juego.

Otro aspecto importante que es necesario resaltar, es que el 66% de lo(a)s estudiantes se enfocó en terminar directamente el juego, mientras que el otro 33% recorrió gran parte de la ciudad e interactuó con diferentes objetos y NPC's antes de realizar las misiones. Además, también se apreció que la mayoría de lo(a)s estudiantes no tuvo la curiosidad de probar los elementos de la interfaz como lo son los íconos para consultar el resumen de notas, misión actual, tienda y mapa de *stakeholders*, los cuales se encuentran en la parte superior de la interfaz, ya que se sintieron más atraídos hacia elementos del escenario como animales, coches, y sonidos. Por lo tanto, fue necesario que la tesista interviniera para explicar que era fundamental que usarán tales elementos puesto que son necesarios para realizar las misiones del juego. Esta situación evidenció que es necesario mejorar el JS incluyendo misiones cortas en el tutorial con el fin de que lo(a)s estudiantes se relacionen con la importancia de estos elementos del juego y optimicen su experiencia durante la realización de las misiones.

Aunado a lo anterior, se observó que aproximadamente a los 10 minutos de haber iniciado la sesión de juego, la mayoría de lo(a)s estudiantes comenzó a familiarizarse con el juego. Sin embargo, dos estudiantes se mostraron un poco más lentos puesto que se les dificultó entender la jugabilidad

del juego y se notaban un poco frustrados debido a que sus compañero(a)s avanzaban más rápido en la partida. No obstante, conforme pasó el tiempo, esto(a)s estudiantes lograron sentirse cómodos y encontraron la forma de terminar las misiones.

El(la) estudiante que más rápido terminó la sesión de juego lo hizo en un tiempo de 35 minutos, mientras que el(la) más lento(a) lo hizo en 75 minutos, superando el tiempo límite establecido. Esta situación se atribuyó directamente a la falta de experiencia que el(la) estudiante tenía con los videojuegos.



Figura 4.3. Estudiantes del grupo experimental realizando el tutorial del juego

Como se explicó anteriormente, el juego presenta tres diferentes finales posibles: dos finales en donde el(la) estudiante pierde, ya sea por identificar incorrectamente a los *stakeholders* o por

identificarlos correctamente pero perder el combate final, y otro donde el(la) estudiante gana al identificar correctamente a los *stakeholders*. En este sentido, el 83.33% de los estudiantes completó satisfactoriamente el juego y únicamente el 16.67% falló al momento de identificar correctamente a los *stakeholders*. Esta falta de éxito se puede atribuir a la falta de atención que pusieron esto(a)s estudiantes a la información proporcionada por los *stakeholders* durante la sesión de juego y al no utilizar el ícono de ayuda para recordar los diálogos e información de la misión.

Al finalizar la sesión de juego, lo(a)s estudiantes del grupo experimental respondieron el cuestionario de evaluación de desempeño académico y, posteriormente, proporcionaron sus percepciones a los planteamientos hechos por el instrumento de evaluación de MEEGA+.



Figura 4.4. Estudiantes del grupo experimental durante la sesión de juego

Posteriormente se procedió a analizar las respuestas de lo(a)s estudiantes del grupo experimental para obtener que el 8% obtuvo una puntuación de 9 aciertos correctos de 15, el 25% logró responder correctamente 11 aciertos de 15, el 16% mostró 12 aciertos correctos de 15, el 17% alcanzó a responder correctamente 13 aciertos de 15, el 17% de los estudiantes obtuvo 14 aciertos de 15, y el 17% realizó un trabajo excelente al obtener la totalidad de los aciertos (ver Figura 4.5). Esta información permitió calcular una calificación promedio de 8.0/10, mejorando sustancialmente la calificación obtenida por el grupo de control que fue de 6.7/10.

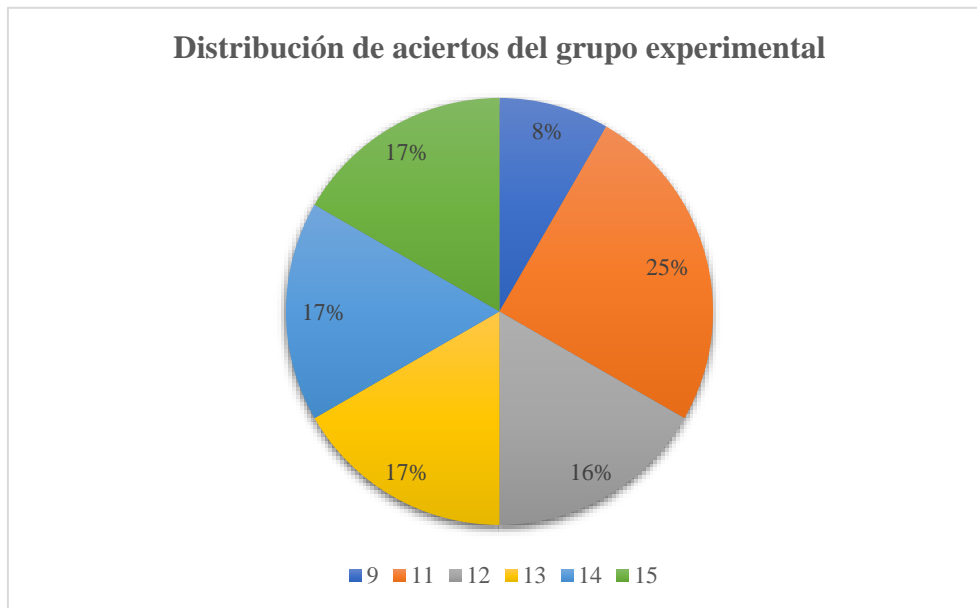


Figura 4.5. Evaluación del desempeño académico correspondiente al grupo experimental

Como es posible apreciar en la Figura 4.5, el 17% de los alumnos del grupo experimental tuvo una puntuación perfecta (color verde). Además, considerando los resultados del grupo de control, aumentó la cantidad de estudiantes que obtuvieron un mejor desempeño con más de 10 aciertos correctos (92%), ya que en el grupo de control únicamente el 17% de lo(a)s estudiantes obtuvo puntuaciones en este rango. En este sentido, queda claro que el grupo experimental mostró un mejor resultado académico que el grupo de control que siguió un enfoque tradicional de enseñanza/aprendizaje.

Por lo tanto, a partir de los resultados obtenidos, se puede inferir que hacer uso de métodos no tradicionales en la enseñanza/aprendizaje de la identificación correcta de los *stakeholders* mejora la asimilación de conceptos, puesto que un JS pueden involucrar elementos de aprendizaje que les resulten atractivos a los estudiantes universitarios (e.g., gráficos, personajes, escenarios, etc.).

4.4.3. Evaluación de la efectividad del juego

La segunda evaluación de “*Misterio en Wonderland*” consistió en la aplicación del instrumento de medición de MEEGA+ al grupo experimental. El objetivo fue determinar la calidad del juego en términos de motivación, experiencia de lo(a)s estudiantes y aprendizaje. Así pues, el modelo MEEGA+ fue desarrollado sistemáticamente a través de la descomposición de los objetivos de evaluación en medidas hasta llegar a un instrumento de medición que evalúa la calidad percibida por los estudiantes sobre un juego educativo en términos nueve dimensiones: usabilidad, confianza, desafío, satisfacción, interacción social, diversión, atención enfocada, relevancia, y aprendizaje

percibido. Los resultados obtenidos con MEEGA+ contribuirán a mejorar “*Misterio en Wonderland*” en el futuro para lograr una adopción efectiva y eficiente en la práctica real.

Cabe enfatizar que, como se indicó anteriormente, “*Misterio en Wonderland*” no fue creado como un juego multijugador, por lo cual la dimensión de interacción social no fue incluida en la evaluación. A continuación se presentan los resultados obtenidos para cada dimensión evaluada.

4.4.3.1. Evaluación de la usabilidad del juego

El instrumento de medición de MEEGA+ define 12 ítems para recoger las percepciones de los estudiantes sobre la usabilidad de un juego. En este sentido, el instrumento establece la necesidad de evaluar cinco subdimensiones para determinar la usabilidad, las cuales son estética, capacidad de aprendizaje, operabilidad, accesibilidad y protección al usuario de posibles errores. La Tabla 5 muestran las percepciones de los estudiantes en este sentido.

Tabla 5. Percepciones de los estudiantes sobre la usabilidad del juego

Dimensión por evaluar		Ítem	TD	ED	N	DA	TA	M	SD
Usabilidad	Estética	1. El diseño del juego me pareció atractivo, respecto a la interfaz, gráficos, etc.	2	1	1	6	2	0.42	1.32
		2. Las fuentes de texto y los colores me parecieron adecuados y consistentes.	2	1	1	5	3	0.50	1.38
		3. Necesité usar el tutorial para comprender la temática del juego.	1	1	2	4	4	0.75	1.23
	Capacidad de aprendizaje	4. Interpreté de forma fácil la jugabilidad.	3	0	2	4	3	0.33	1.49
		5. Considero que la mayoría de personas puede aprender los comandos del juego de manera rápida.	2	1	0	5	4	0.67	1.43
	Operabilidad	6. Pude jugar de forma fácil.	1	1	6	4	0	0.08	0.86
		7. Las reglas del juego me parecieron claras y concisas.	2	1	1	5	3	0.50	1.38
	Accesibilidad	8. El tamaño y estilo de las fuentes me fueron fáciles de leer.	1	0	0	5	6	1.25	1.09
		9. Considero que colores fueron adecuados para la temática del juego.	0	0	0	5	7	1.58	0.49
		10. Me fue posible personalizar la apariencia del juego (fuentes, colores, etc.) según mis preferencias.	0	3	4	3	2	0.33	1.03
	Protección al usuario de posibles errores	11. El juego me impidió cometer errores.	0	2	5	4	1	0.33	0.85
		12. Cuando cometí un error, me fue fácil recuperarme.	0	3	0	7	2	0.67	1.03

TD= Totalmente en desacuerdo, **ED** = En desacuerdo, **N**= Neutral, **DA**= De acuerdo, **TA**= Totalmente de acuerdo, **M**= Media aritmética, **SD**= Desviación estándar.

De acuerdo con las respuestas obtenidas, y considerando el cálculo de la media y la desviación estándar, se obtuvieron las siguientes conclusiones:

- **Estética:** A pesar de que se presentó una disparidad considerable en las respuestas de lo(a)s estudiantes para esta subdimensión ($SD= 1.20$), la mayoría de esto(a)s coincidieron en que los elementos estéticos del JS fueron importantes para motivarlos a jugar ($M= 0.55$). De hecho, las percepciones sobre el tutorial evidenciaron la situación que se explicó en los resultados del grupo experimental, puesto que el 33.33% de lo(a)s estudiantes no consideró importante el prestar atención al tutorial del juego (*ítem 3. Necesité usar el tutorial para comprender la temática del juego*).
- **Capacidad de aprendizaje:** De nuevo, la disparidad en la percepciones de lo(a)s estudiantes evidenció sus diferentes valoraciones después de jugar ($SD= 1.20$). No obstante, la mayoría de los estudiantes consideró que el diseño del juego facilitó su uso ($M= 0.50$). En este sentido, el 75% de los estudiantes consideró que la mayoría de las personas podrían usarlo fácilmente (*ítem 5. Considero que la mayoría de personas puede aprender los comandos del juego de manera rápida*).
- **Operabilidad:** La disparidad en las respuestas de los estudiantes se redujo ($SD= 1.12$), lo que deja en claro que estuvieron más de acuerdo en sus valoraciones. Sin embargo, se observó una percepción generalizada sobre la dificultad que implicó el realizar las misiones durante la sesión de juego ($M= 0.08$, $SD= 0.86$), puesto que el 66.66% de lo(a)s estudiantes no respondió afirmativamente (*ítem 6. Pude jugar de forma fácil*). Como se mencionó anteriormente, el principal motivo que causó esta percepción es la falta de experiencia previa de alguno(a)s estudiantes con los videojuegos. Por otro lado, el 66.66% consideró que las reglas del JS fueron claras y concisas durante la sesión de juego ($M= 0.50$).
- **Accesibilidad:** Lo(a)s estudiantes consideraron que los elementos gráficos del JS, como tamaño y estilo de las letras, los colores, y la apariencia, fueron los adecuados para facilitar la accesibilidad a las principales funcionalidades del mismo ($M= 0.50$, $SD= 0.87$). De hecho, el 100% de lo(a)s estudiantes coincidió en que los colores utilizados encajaron perfectamente con la temática reflejada en el JS (*ítem 9. Considero que colores fueron adecuados para la temática del juego*).
- **Protección al usuario de posibles errores:** Finalmente, lo(a)s estudiantes coincidieron de manera general en que fueron capaces de retomar la realización de las misiones cuando cometieron algún error ($M= 0.67$, $SD= 1.03$). Sin embargo, la mayoría coincidió en que el juego no les impidió cometer tales errores y, por consiguiente, su percepción tendió a ser negativa. En este sentido, es importante mencionar que esta información permitió detectar que el JS debe ser mejorado para incluir mensajes o acciones de alerta que prevengan al(a) estudiante de realizar una acción incorrecta.

4.4.3.2. Evaluación de la confianza de lo(a)s estudiantes

De acuerdo con el modelo MEEGA+, las percepciones de los estudiantes sobre la confianza puesta en el juego permiten evaluar su capacidad de avanzar en el aprendizaje de contenidos educativos a través de su esfuerzo y habilidad (e.g., a través de tareas con un nivel de dificultad creciente). Considerando lo anterior, la Tabla 6 muestra los resultados obtenidos con la aplicación del instrumento de medición de MEEGA+ a los estudiantes del grupo experimental.

Tabla 6. Percepciones de los estudiantes sobre la confianza en el juego

Dimensión por evaluar	Ítem	TD	ED	N	DA	TA	M	SD
Confianza	13. Desde el inicio del juego me dio la impresión de que iba a ser fácil de jugar.	0	0	0	10	2	1.17	0.37
	14. El contenido y la estructura del juego me dieron la confianza de que aprendería algo nuevo.	0	0	2	7	3	1.08	0.64

TD= Totalmente en desacuerdo, ED = En desacuerdo, N= Neutral, DA= De acuerdo, TA= Totalmente de acuerdo, M= Media aritmética, SD= Desviación estándar.

Las respuestas de lo(a)s estudiantes evidenciaron que experimentaron altos niveles de confianza al utilizar un instrumento alternativo de aprendizaje, como lo es el JS propuesto en esta Tesis. Como se puede observar, desde su primer contacto con el juego tuvieron la sensación de que iba a ser fácil de jugar (M= 1.17, SD= 0.37) y, además, estaban seguros de que aprenderían algo nuevo (M= 1.08, SD= 0.64). En este sentido, se reforzó la comprensión de los conceptos expuestos en la clase teórica. Cabe mencionar que la confianza fue la dimensión que mejor fue evaluada por lo(a)s estudiantes, además de que se alcanzó el nivel más bajo de disparidad en las respuestas para todas las dimensiones.

4.4.3.3. Evaluación del desafío del juego

Esta dimensión pretende evaluar en qué medida el JS es suficientemente desafiante con respecto al nivel de competencia de lo(a)s estudiantes. Es decir, el modelo MEEGA+ indica que el aumento de la dificultad debe ocurrir a un ritmo apropiado acompañando a la curva de aprendizaje. Por lo tanto, es necesario presentar nuevos obstáculos y situaciones a lo largo del juego para minimizar la fatiga y mantener el interés de lo(a)s estudiantes. Así pues, la Tabla 7 muestra las percepciones del grupo experimental con relación al desafío presentado por “*Misterio en Wonderland*”.

Tabla 7. Percepciones de los estudiantes sobre el desafío del juego

Dimensión por evaluar	Ítem	TD	ED	N	DA	TA	M	SD
Desafío	15. Considero que la dificultad del juego fue la apropiada.	0	0	1	8	3	1.17	0.55
	16. Considero que el ritmo de las misiones (situaciones, variaciones, nuevos obstáculos, etc.) iba variando adecuadamente conforme el tiempo avanzaba.	0	1	0	8	3	1.08	0.76
	17. Considero que el juego y las misiones no se vuelven monótonas o aburridas a medida que avanza la historia.	0	0	4	5	3	0.92	0.76

TD= Totalmente en desacuerdo, ED = En desacuerdo, N= Neutral, DA= De acuerdo, TA= Totalmente de acuerdo, M= Media aritmética, SD= Desviación estándar.

De acuerdo con las respuestas de lo(a)s estudiantes, la inmensa mayoría consideró que la dificultad propuesta por el JS fue la correcta para abordar el tópico de identificación de los *stakeholders* (M= 1.17, SD= 0.55). Por otro lado, un(a) estudiante consideró que el ritmo en que se

desarrollaron las misiones no fue el correcto ($M= 1.08$, $SD= 0.76$). Al indagar sobre su respuesta, el(la) estudiante argumentó que su percepción se debió a que, dada su falta de experiencia con los videojuegos, prácticamente tardó en terminar todas las misiones y sintió presión por finalizar la sesión de juego junto con sus compañero(a)s.

Finalmente, el 33.33% de lo(a)s estudiantes se mantuvieron neutrales en cuanto a evaluar la monotoneidad de las misiones (ítem 17. *Considero que el juego y las misiones no se vuelven monótonas o aburridas a medida que avanza la historia*). A pesar de que la mayoría de ésto(a)s consideró que no se les hizo un juego con misiones tediosas y/o aburridas, se decidió indagar con los cuatro estudiantes que proporcionaron valoraciones neutrales. De manera similar que en la evaluación de la dimensión de desafío, lo(a)s estudiantes argumentaron tener poca experiencia con el uso de los videojuegos, por lo que la dificultad de utilizar el teclado y ratón de la computadora para tomar objetos o pelear con villanos hizo que dieran esa respuesta.

4.4.3.4. Evaluación de la satisfacción de lo(a)s estudiantes

El modelo MEEGA+ establece que la satisfacción de los estudiantes se relaciona con la sensación que experimentan al darse cuenta de que el esfuerzo realizado en el juego se ve reflejado en su aprendizaje. En este sentido, la Tabla 8 resume la información proporcionada por los estudiantes al respecto.

Tabla 8. Percepciones de los estudiantes sobre la satisfacción con el juego

Dimensión por evaluar	Ítem	TD	ED	N	DA	TA	M	SD
Satisfacción	18. El completar cada una de las misiones del juego me daba una sensación de logro.	0	0	1	5	6	1.42	0.64
	19. Gracias a mi esfuerzo pude sortear los obstáculos del juego e ir avanzando.	0	0	1	8	3	1.17	0.55
	20. Me siento satisfecho(a) con el aprendizaje obtenido.	0	0	1	7	4	1.25	0.60
	21. Recomendaría este juego a mis amigo(a)s o compañero(a)s.	0	0	1	6	5	1.33	0.62

TD= Totalmente en desacuerdo, ED = En desacuerdo, N= Neutral, DA= De acuerdo, TA= Totalmente de acuerdo, M= Media aritmética, SD= Desviación estándar.

Es importante mencionar que la dimensión de satisfacción fue la segunda mejor evaluada por lo(a)s estudiantes, puesto que la media alcanzó los valores más altos y la desviación estándar se mantuvo baja para todos los ítems. Concretamente, el 92% de lo(a)s estudiantes consideró que su esfuerzo les permitió concluir las misiones dentro del juego y les proporcionó, como consecuencia, satisfacción y una sensación de logro personal. Aunado a esto, lo(a)s estudiantes coincidieron en que podrían recomendar el uso de “*Misterio en Wonderland*” a sus amigo(a)s y/o compañero(a)s que quisieran reforzar sus conocimientos sobre IR y la identificación de los *stakeholders*.

4.4.3.5. Evaluación de la diversión dentro del juego

Para el modelo MEEGA+, la diversión debe evaluarse en términos de las sensaciones de placer, felicidad, relajación y distracción que experimentan los estudiantes durante el uso del juego creado para complementar el aprendizaje. La Tabla 9 muestra la información recogida sobre dicha dimensión.

Tabla 9. Percepciones de los estudiantes sobre la diversión del juego

Dimensión por evaluar	Ítem	TD	ED	N	DA	TA	M	SD
Diversión	22. Me divertí con el juego.	0	0	0	7	5	1.42	0.49
	23. Algunos elementos del juego (diálogos, misiones, videos, etc.) me hicieron reír.	0	0	1	4	7	1.50	0.65

TD= Totalmente en desacuerdo, ED = En desacuerdo, N= Neutral, DA= De acuerdo, TA= Totalmente de acuerdo, M= Media aritmética, SD= Desviación estándar.

En términos generales, la percepción de diversión de lo(a)s estudiantes fue positiva (*ítem 22. Me divertí con el juego*, M= 1.42, SD= 0.49). Además, se observó cierto grado de alegría generalizada puesto que los estudiantes se mostraron sonrientes en los puntos del juego donde se reproducían cinemáticas, descubrían pistas, o personalizaban a su avatar con diferentes accesorios o mascotas (*ítem 23. Algunos elementos del juego me hicieron reír*, M= 1.50, SD= 0.65).

Es importante mencionar que, durante el diseño y posterior desarrollo del juego, se agregaron algunos elementos al JS cuya función principal era hacer más amena la interacción entre el estudiante y el método de aprendizaje y, como se puede observar con la información recogida para esta dimensión, se logró el cometido.

4.4.3.6. Evaluación de la atención enfocada de lo(a)s estudiantes

MEEGA+ argumenta que la evaluación de la atención enfocada de los estudiantes debe evaluarse considerando la atención, concentración, absorción y disociación temporal experimentadas durante las sesiones de juego. Con este objetivo en mente, la Tabla 10 presenta la información proporcionada por lo(a)s estudiantes después de usar “*Misterio en Wonderland*”.

Tabla 10. Percepciones de los estudiantes sobre su atención enfocada en el juego

Dimensión por evaluar	Ítem	TD	ED	N	DA	TA	M	SD
Atención enfocada	24. Hubo algo que llamó mi atención desde el inicio del juego.	0	0	3	4	5	1.17	0.80
	25. Estaba tan concentrado(a) en las misiones del juego que perdí la noción del tiempo.	0	1	3	2	6	1.08	1.04
	26. Me olvidé de mi entorno mientras jugaba.	0	1	1	3	7	1.33	0.94

TD= Totalmente en desacuerdo, ED = En desacuerdo, N= Neutral, DA= De acuerdo, TA= Totalmente de acuerdo, M= Media aritmética, SD= Desviación estándar.

De acuerdo con las opiniones de los estudiantes, el 75% de ello(a)s coincidió en que el juego fue de su interés desde un inicio y el 25% se mantuvo neutral (*ítem 24. Hubo algo que llamó mi atención desde el inicio del juego*). Al indagar al respecto con lo(a)s estudiantes, se pudo comprobar que lo(a)s mismo(a)s estudiantes que no tenían experiencia previa con el uso de videojuegos, son los que no se sintieron atraído(a)s al inicio para interactuar con el JS. Algo similar ocurrió cuando el 8.33% de lo(a)s estudiantes argumentó no sentirse inmerso en el juego (*ítem 25. Estaba tan concentrado(a) en las misiones del juego que perdí la noción del tiempo*, M= 1.08, SD= 1.04) ni haber experimentado disociación temporal al realizar las misiones (*ítem 26. Me olvidé de mi entorno mientras jugaba*, M= 1.33, SD= 0.94). No obstante, un promedio del 75% de lo(a)s estudiantes

coincidió en haber experimentado niveles altos de atención durante el esclarecimiento del asesinato en “*Misterio en Wonderland*”.

4.4.3.7. Evaluación de la relevancia del juego

La relevancia del juego, de acuerdo con MEEGA+, se determina evaluando la capacidad de lo(a)s estudiantes de darse cuenta de que la propuesta educativa es coherente con sus objetivos y que es posible vincular los contenidos explorados con su futuro académico o profesional. En este sentido, la Tabla 11 resume las opiniones vertidas por lo(a)s estudiantes al respecto.

Tabla 11. Percepciones de los estudiantes sobre la relevancia del juego

Dimensión por evaluar	Ítem	TD	ED	N	DA	TA	M	SD
Relevancia	27. Considero que los contenidos del juego fueron relevantes.	0	0	2	6	4	1.17	0.69
	28. Comprendí la relación del juego con los objetivos de aprendizaje.	0	0	3	6	3	1.00	0.71
	29. El juego es una forma adecuada de proporcionar aprendizaje sobre el tema.	0	0	0	8	4	1.33	0.47
	30. Prefiero aprender con este juego que con otras formas (libros, diapositivas, etc.).	0	0	3	5	4	1.08	0.76

TD= Totalmente en desacuerdo, ED = En desacuerdo, N= Neutral, DA= De acuerdo, TA= Totalmente de acuerdo, M= Media aritmética, SD= Desviación estándar.

La información presentada en esta tabla muestra que el 100% de lo(a)s estudiantes coincidió en que “*Misterio en Wonderland*” representa una alternativa útil para complementar el aprendizaje sobre la identificación de los *stakeholders* (ítem 29. *El juego es una forma adecuada de proporcionar aprendizaje sobre el tema*, M= 1.33, SD= 0.47). Por otro lado, tres estudiantes permanecieron neutrales cuando se les preguntó si habían comprendido la relación entre el juego y los objetivos de aprendizaje establecidos en la clase teórica (i.e., actividades para la identificación de los *stakeholders*) (M= 1.00, SD= 0.71) y si preferían utilizar el juego para reforzar la comprensión de los conceptos teóricos sobre IR en lugar de herramientas tradicionales (M= 1.08, SD= 0.76). La explicación de sus respuestas neutrales se debió, una vez más, a la nula experiencia que tenían jugando videojuegos. No obstante, esta información también se considera importante para mejorar el JS con el objetivo de mejorar su jugabilidad para estudiantes con esta característica. Finalmente, el 83.33% de lo(a)s estudiantes consideró que los elementos mostrados en el juego fueron relevantes para comprender mejor los conceptos teóricos aprendidos en el aula (M= 1.17, SD= 0.69).

4.4.3.8. Evaluación del aprendizaje percibido por lo(a)s estudiantes

El modelo MEEGA+ establece que el factor de calidad relacionado con el aprendizaje percibido por lo(a)s estudiantes, sea evaluado considerando dos dimensiones: el aprendizaje a corto plazo y los objetivos de aprendizaje. La evaluación del aprendizaje a corto plazo se basa en el modelo de evaluación de Moody y Sindre (2003) y se considera el efecto general del juego en el aprendizaje de lo(a)s estudiantes durante el curso. En este sentido, la Tabla 12 muestra los resultados obtenidos de aplicar el instrumento de medición para recoger las percepciones de lo(a)s estudiantes del grupo experimental. La información mostrada en la tabla corresponde con el aprendizaje percibido a corto plazo.

Tabla 12. Percepciones de los estudiantes sobre el aprendizaje experimentado con el juego

Dimensión por evaluar	Ítem	TD	ED	N	DA	TA	M	SD
Aprendizaje percibido (a corto plazo)	31. El juego contribuyó a mejorar el aprendizaje del tópico visto en clase.	0	0	2	8	2	1.00	0.58
	32. La enseñanza brindada por el juego fue más eficiente para el aprendizaje del tópico visto en clase que otros métodos tradicionales.	0	1	1	9	1	0.83	0.69

TD= Totalmente en desacuerdo, ED = En desacuerdo, N= Neutral, DA= De acuerdo, TA= Totalmente de acuerdo, M= Media aritmética, SD= Desviación estándar.

De acuerdo con la información recogida, el 83.33% de lo(a)s estudiantes consideró que “*Misterio en Wonderland*” los ayudó a mejorar la comprensión de los conceptos teóricos sobre la identificación de los *stakeholders* (M= 1.00, SD= 0.58), incluso el mismo porcentaje de estudiantes consideró que el juego fue más eficiente que los métodos tradicionales de enseñanza/aprendizaje (M= 0.83, SD= 0.69). Es importante mencionar que, a pesar de estos resultados, lo(a)s estudiantes con nula experiencia con los videojuegos continuaron dando valoraciones negativas y, por lo tanto, sus percepciones darán lugar a mejorar la jugabilidad del JS.

Por otro lado, el modelo MEEGA+ indica que para realizar la evaluación de la cobertura sobre los objetivos de aprendizaje, el instrumento de medición debe de personalizarse para determinar precisamente el aprendizaje percibido por lo(a)s estudiantes, sobre tales objetivos (ver Tabla 2), durante las sesiones de juego. En este sentido, los juegos enfocados a cubrir alguna de las áreas de las Ciencias de la Computación regularmente se utilizan para mejorar el conocimiento en los niveles cognitivos de recuerdo, comprensión y aplicación de los conocimientos, de acuerdo con la Taxonomía de Bloom (Krathwohl, 2002). Así pues, la Tabla 13 muestra la información recogida al respecto.

Tabla 13. Percepciones de los estudiantes sobre la cobertura de los objetivos de aprendizaje

Dimensión por evaluar	Ítem	TD	ED	N	DA	TA	M	SD
Aprendizaje percibido (objetivos de aprendizaje)	33. El juego contribuyó a mejorar mi comprensión sobre cómo identificar a los <i>stakeholders</i> .	0	0	1	7	4	1.25	0.60
	34. El juego contribuyó a ampliar mi entendimiento sobre las clases de usuarios que pueden representar a los <i>stakeholders</i> .	0	0	0	1	11	1.92	0.28
	35. El juego contribuyó a que aplicara correctamente la teoría para interactuar correctamente con los posibles <i>stakeholders</i> .	0	0	2	9	1	0.92	0.49
	36. El juego contribuyó a que aprendiera cómo utilizar un esquema para caracterizar y evaluar las relaciones entre los <i>stakeholders</i> identificados.	0	0	0	10	2	1.17	0.37

TD= Totalmente en desacuerdo, ED = En desacuerdo, N= Neutral, DA= De acuerdo, TA= Totalmente de acuerdo, M= Media aritmética, SD= Desviación estándar.

Las percepciones de lo(a)s estudiantes para esta evaluación adicional de la cobertura que “*Misterio en Wonderland*” hizo sobre los objetivos de aprendizaje fue casi excelente, puesto que el 93.75% de los integrantes del grupo experimental proporcionó valoraciones positivas.

4.4.3.9. Consideraciones finales sobre los resultados obtenidos en la evaluación empírica

Considerando las secciones anteriores, se observó que el grupo de control obtuvo una calificación promedio de 6.7/10 (i.e., un total de 10/15 aciertos) mientras que el grupo experimental obtuvo una calificación promedio de 8/10 (i.e., un total de 12/15 aciertos), lo que permitió deducir que el uso de “*Misterio en Wonderland*” es útil como un complemento educativo, puesto que cumplió el propósito de mejorar la enseñanza/aprendizaje sobre la identificación de los *stakeholders* a nivel licenciatura, en comparación con una clase presencial tradicional.

En este sentido, de acuerdo con las opiniones proporcionadas por lo(a)s integrantes del grupo experimental, se puede afirmar que el juego demostró calidad en términos de experiencia y aprendizaje percibido (ver Tablas 5 a 13). Además, el número de estudiantes cuya calificación era mayor a 10/15 puntos se aumentó considerablemente con las sesiones de juego, inclusive, hubo estudiantes con puntuaciones perfectas de 15/15. Así mismo, la mayoría de lo(a)s estudiantes que conformaron al grupo experimental afirmó que “*Misterio en Wonderland*” contribuyó a mejorar su comprensión sobre los conceptos teóricos sobre la identificación de los *stakeholders*. Por tales motivos, se puede concluir que, en el contexto de la presente evaluación empírica, el JS propuesto en esta Tesis cumplió el cometido de proporcionar un entorno seguro donde los estudiantes aprendan y se diviertan, sin que sepan necesariamente que se están repasando conceptos sobre la identificación de los *stakeholders* que podrían ser difíciles de entender únicamente con clases teóricas.

Por lo tanto, y tomando en consideración la información mostrada en este capítulo de la Tesis, se ha aportado evidencia suficiente para aceptar la hipótesis planteada en el Capítulo 1 bajo las particularidades de la evaluación empírica conducida. Es importante mencionar que los resultados y las conclusiones aquí mostradas no pueden generalizarse, puesto que esto requeriría el diseño y realización de una mayor cantidad de casos de estudio.

4.4.3.10. Recomendaciones y sugerencias del grupo experimental para mejorar el juego

Por otro lado, además de los resultados mostrados, al final de la evaluación empírica se recogió información que permitió configurar una serie de recomendaciones y sugerencias proporcionadas por lo(a)s estudiantes del grupo experimental que buscan favorecer la jugabilidad del juego y mejorar la transferencia de información durante las sesiones de juego. Esta información se resume en la Tabla 14.

Tabla 14. Recomendaciones y sugerencias de los estudiantes del grupo experimental respecto al diseño del juego

Recomendación	Comentario de la tesista
<p>“Sugiero establecer una velocidad menor para la movilidad del personaje, ya que actualmente no se le permite realizar movimientos pequeños”.</p>	<p>Al ser un juego en 3D, es necesario usar equipos que tengan las características de operación necesarias para ejecutarlo correctamente. En caso contrario, la cantidad de FPS (Fotogramas Por Segundo) a los que el juego se ejecuta aumentaría, provocando problemas durante el desface de los personajes. En este sentido, algunas de las computadoras del laboratorio donde el juego fue probado no cumplían con tales características, lo que ocasionó este tipo de problemas. No obstante, se prestará mayor atención en las evaluaciones futuras para evitar que este problema se repita.</p>

<p><i>“Me gustaría que se aplicarán animaciones al caminar hacia los lados y hacia atrás”.</i></p>	<p>Debido a que se trata de un juego en tercera persona donde la movilidad depende tanto del teclado como del ratón de la computadora, la animación de algunos movimientos no es estéticamente visual, por tal motivo, se implementaron movimientos únicamente para cuando el personaje camina hacia adelante. No obstante, este detalle poder ser mejorado en una versión futura del juego.</p>
<p><i>“Considero que se podría impedir el movimiento del personaje mientras se realizan acciones específicas, por ejemplo cuando se interactúa con el inventario”.</i></p>	<p>Durante la creación del juego se detectaron un par de acciones que impedían su funcionamiento correcto y que podían provocar errores de los cuales sería imposible recuperarse, una de ellas es precisamente ésta. Por tal motivo, se tomaron decisiones de diseño que llevaron a permitir que el personaje se siguiera moviendo mientras se interactuaba con el inventario, con el fin de no afectar la jugabilidad. No obstante, se puede investigar más a fondo como implementar una solución que sea más agradable para lo(a)s estudiantes.</p>
<p><i>“El mini mapa que me ayudó a orientarme durante la sesión de juego, no muestra una brújula que me guie hacia la dirección a donde se debía llegar y en algunas ocasiones se me complicaba encontrar la dirección correcta. Por lo tanto, me gustaría que se agregará este elemento al juego”.</i></p>	<p>A pesar de que el juego ya muestra una línea roja que se debe seguir para llegar a la ubicación donde se realizaría la misión, alguno(a)s estudiantes mencionaron que no era suficiente, ya que al ser muy grande el mapa es fácil perderse y una vez ubicada la línea roja, no se sabía que dirección se debía que seguir. Sin duda alguna que esta sugerencia mejoraría la jugabilidad y, por ende, se implementará en una versión mejorada del juego.</p>
<p><i>“Creo que el contador de tiempo debería de ser más visible, así mismo debería mostrar alguna alerta cuando el tiempo está por agotarse”.</i></p>	<p>Esta sugerencia de lo(a)s estudiantes es apropiada y se considera que podría también ayudar a mejorar la jugabilidad y experiencia de los estudiantes durante las sesiones de juego, por lo que se tomará en cuenta en la mejora del mismo.</p>
<p><i>“El rango de visibilidad del jugador para hablar con una persona debería ser más grande, es decir, se debe poder hablar con el personaje cuando se está al lado de él y no solo cuando se está enfrente de los personajes”.</i></p>	<p>Esta sugerencia de lo(a)s estudiantes es apropiada y se considera que podría también ayudar a mejorar la jugabilidad y experiencia de los estudiantes durante las sesiones de juego, por lo que se tomará en cuenta en la mejora del mismo.</p>
<p><i>“Creo que es importante que se disminuya el volumen del sonido del entorno cuando se habla con los personajes del juego”.</i></p>	<p>Esta sugerencia de lo(a)s estudiantes es apropiada y se considera que podría también ayudar a mejorar la jugabilidad y experiencia de los estudiantes durante las sesiones de juego, por lo que se tomará en cuenta en la mejora del mismo.</p>
<p><i>“Sugiero poner más distancia entre los personajes al momento del combate, ya que parece que los personajes se sobreponen”.</i></p>	<p>Esta sugerencia de lo(a)s estudiantes es apropiada y se considera que podría también ayudar a mejorar la jugabilidad y experiencia de los estudiantes durante las sesiones de juego, por lo que se tomará en cuenta en la mejora del mismo.</p>
<p><i>“Se sugiere hacer uso de una vista isométrica del juego, ya que es fácil perderse dentro de un mundo 3D, con la vista isométrica, es más fácil orientarse y se evita el uso de mapas”.</i></p>	<p>Esta sugerencia de lo(a)s estudiantes es apropiada y se considera que podría también ayudar a mejorar la jugabilidad y experiencia de los estudiantes durante las sesiones de juego, por lo que se tomará en cuenta en la mejora del mismo.</p>

5. Conclusiones

Durante el desarrollo de este trabajo de Tesis se puso en evidencia la deficiencia de conocimientos que tienen lo(a)s estudiantes de pregrado o licenciatura en lo que respecta al conocimiento sobre la elicitación de requisitos, específicamente en la identificación correcta de los *stakeholders*. Esta situación se complica cuando, para su enseñanza, se sigue únicamente un método tradicional, ya que muchas veces lo(a)s profesores no logran transmitir correctamente la información a lo(a)s estudiantes, volviendo esta acción tediosa y, como consecuencia, les resulta difícil mantener la atención durante la clase con el fin de retener la mayor cantidad de información posible. En este sentido, dado que la IS es un campo de estudio donde es imprescindible la comprensión y el entendimiento de los conceptos teóricos para usarlos correctamente en la práctica, su aprendizaje puede resultar tedioso y aburrido para lo(a)s estudiantes. Por tal motivo, es necesario la búsqueda de métodos alternativos de aprendizaje para complementar los enfoques tradicionales de enseñanza en pro de aminorar estos problemas.

Por lo tanto, el presente documento de Tesis expuso la existencia de métodos alternativos de enseñanza que pueden ser útiles como complemento de los tradicionales para promover, de manera efectiva, la transferencia de conocimientos a nivel licenciatura en la educación sobre IR. En concreto, los juegos serios. Este tipo de herramientas educativas facilita la acción de que el(la) estudiante adquiera información a la vez que se divierte, dando como consecuencia que puedan adquirir el conocimiento que necesitarán en la industria de desarrollo de *software* al complementar experiencias didácticas con los conocimientos adquiridos en las aulas. En este sentido, este trabajo de Tesis propuso el diseño, creación y uso del juego serio llamado “*Misterio en Wonderland*”, el cual proporciona un entorno virtual seguro para que lo(a)s estudiantes realicen misiones con el fin de reforzar los conocimientos teóricos relacionados con la identificación, exacta y correcta, de los *stakeholders* en la elicitación de los requisitos de *software*. Con este objetivo en mente, se presentaron los resultados obtenidos mediante la realización de un caso de estudio que involucró a estudiantes reales de la Universidad Tecnológica de la Mixteca, quienes reforzaron su aprendizaje mediante sesiones de trabajo con dicho juego. Como se demostró en el Capítulo 4 de este documento, el juego creado complementó eficientemente la enseñanza/aprendizaje de un tópico de la IR, puesto que lo(a)s estudiantes que formaron parte del grupo experimental obtuvieron un mejor desempeño académico en comparación con el obtenido por los estudiantes del grupo de control. Aunado a esto, se ha proporcionado evidencia de la efectividad del juego considerando las dimensiones establecidas por el modelo MEEGA+.

Por otro lado, resulta interesante plasmar algunas lecciones aprendidas que se desprenden del diseño e implementación del juego presentado en el Capítulo 3 de esta Tesis. Por ejemplo, es importante mencionar que el desarrollo de un juego serio digital no es una tarea sencilla, ya que no solamente se deben considerar las habilidades y conocimientos que se tengan sobre programación de

sistemas, sino que es necesario analizar metodologías enfocadas a su concepción y diseño, aspecto que queda fuera del acervo de conocimientos que se adquieren durante la licenciatura.

Así mismo, el establecimiento de la historia del juego resultó un reto importante, ya que esta tarea implica un alto grado de inventiva y creatividad de tal forma que se pueda relacionar dicha historia con objetivos de aprendizaje que aborden el área de la IS que se pretende abordar. Es decir, es complicado pensar en una trama entretenida y, sobre todo, que se preste para que el(la) estudiante refuerce su aprendizaje mientras juega. Aunado a esto, es necesario analizar cuidadosamente qué diálogos deberán incluirse en el juego, puesto que éstos deben ser cortos y concisos, pero deben proporcionar la información suficiente para ubicar al(la) estudiante en el contexto correcto sin aburrirlo o saturarlo con demasiada información.

También es importante enfatizar en la sugerencia de que, cuando se pretenda crear un juego serio digital, lo ideal sería utilizar los *assets* disponibles en la tienda de Unity®, de lo contrario el proceso será más laborioso. En el contexto de “*Misterio en Wonderland*”, por ejemplo, cada elemento de los escenarios se tuvo que agregar uno por uno, como en el caso de las flores, los árboles, los coches, etc., pero hubo algunos elementos que, al no encontrarse dentro de los *assets* de Unity®, fueron creados desde cero, alargando así la terminación de la Tesis.

Por otro lado, se sugiere analizar cuidadosamente la elección de los personajes, puesto que no se trata solamente de escoger y descargar al que sea más agradable visualmente, si no que, dependiendo de las animaciones y movimientos que se desean para el personaje dentro de la historia que se contará con escenas y misiones, se deberán descargar más archivos para posteriormente introducirlos al videojuego, configurarlos y establecer dichos movimientos a base de código y árboles de animaciones dentro de Unity®. Por consiguiente, se sugiere considerar siempre que menos es más.

Finalmente, se sugiere considerar el tiempo que puede llevar el aprender a desarrollar juegos reales cuyo fin sea el apoyar el aprendizaje de cualquier tópico. Es decir, a pesar de que la tesista cursó la materia “Desarrollo de videojuegos” durante los estudios de licenciatura, fue necesario que asistiera a dos diferentes cursos de desarrollo con Unity®, tomar asesorías con personas expertas, así como dedicar tiempo adicional para la lectura de libros especializados. Por lo tanto, se sugiere considerar todas estas actividades en la planificación de la Tesis o proyecto cuyo fin sea la creación de un JS.

Aun con todo el trabajo que implicó el diseño e implementación de “*Misterio en Wonderland*”, es importante asumir desde un inicio que es posible que el videojuego creado no sea del gusto de todas las personas y que se escucharán tanto comentarios positivos como negativos. No obstante, toda la información recibida debe utilizarse para ser mejor en proyectos futuros.

6. Bibliografía

Abdellatif, A., McCollum, B., y McMullan, P. (2018, March). Serious games: Quality characteristics evaluation framework and case study. En *2018 IEEE Integrated STEM Education Conference (ISEC)* (pp. 112-119). IEEE.

Aberkane, A., Poels, G., y Broucke, S. (2021). Exploring automated GDPR-compliance in requirements engineering: A systematic mapping study. *IEEE Access*, 9, 66542-66559.

ACM-CC2020. (2020). Computing Curricula 2020. Recuperado de ACM: <https://www.acm.org/binaries/content/assets/education/curricularecommendations/cc2020.pdf>, 04 de noviembre de 2022.

Almeida, F. (2019). Adoption of a serious game in the developing of emotional intelligence skills. *European Journal of Investigation in Health, Psychology and Education*, 10(1), 30-43.

Anastasiadis, T., Lampropoulos, G., y Siakas, K. (2018). Digital game-based learning and serious games in education. *International Journal of Advances in Scientific Research and Engineering*, 4(12), 139-144.

Bartle, R. (1996). Hearts, clubs, diamonds, spades: Players who suit MUDs. *Journal of MUD research*, 1(1), 19.

Battistella, P., von Wangenheim, C., y Carvalho, O. (2013). Ensinar a gerência de equipes em disciplinas de gerência de projetos de software. *Revista Brasileira de Informática na Educação*, 21(01), 16.

Becker, M., Blanc, R., Casanova, C., Escalante, J., Pralong, L., Ríos, L., y Sosa Zitto, S. (2019). Técnicas para la mejora de la calidad en la ingeniería de requisitos en las empresas de software de Argentina. En *XXI Workshop de Investigadores en Ciencias de la Computación (WICC 2019, Universidad Nacional de San Juan)*.

Berg Marklund, B., Engström, H., Hellkvist, M., y Backlund, P. (2019). What empirically based research tells us about game development. *The Computer Games Journal*, 8, 179-198.

Bilge, P., y Severengiz, M. (2019). Analysis of industrial engineering qualification for the job market. *Procedia Manufacturing*, 33, 725-731.

Blanco, S. M. (2015). Juego educativo enfocado al desarrollo de habilidades para aprender a modelar requerimientos con artefactos UML. Proyecto de investigación. Master en Software Libre. Universidad Abierta de Cataluña.

Bourque, P. y Fairley, R. (2014). SWEBOK: Guide to the software engineering body of knowledge, (Version 3.0). Los Alamitos, CA. IEEE Computer Society.

Brooks, F. (1987). Essence and accidents of software engineering. *IEEE Computer*, 20(4), 10-19.

Bueno, D. (2018). El diseño y el arte en los videojuegos. Proyecto de Fin de Grado. Escuela Universitaria de Informática. Universidad Politécnica de Madrid.

Caeiro-Rodríguez, M., Manso-Vázquez, M., Mikic-Fonte, F. A., Llamas-Nistal, M., Fernández-Iglesias, M. J., Tsalapatas, H., Heidmann, O., De Carvalho, C., Jesmin, T., Terasmaa, J., y Sørensen, L. T. (2021). Teaching soft skills in engineering education: An european perspective. *IEEE Access*, 9, 29222-29242.

Calderón, A., Petri, G., Ruiz, M., y Gresse von Wangenheim, C. (2018). Desarrollando competencias personales y habilidades sociales en ingeniería informática mediante el uso de juegos serios. *Actas de las Jenui*, 3, 127-134.

Calderón, A., Trinidad, M., Ruiz, M., y O'Connor, R. (2019). An experience of use a serious game for teaching software process improvement. En *European Conference on Software Process Improvement* (pp. 249-259). Cham: Springer International Publishing.

Calderón, A., y Ruiz, M. (2016). Uso de juegos serios para la formación en los procesos del ciclo de vida y mejora del software. *Actas de las XXI Jornadas de Ingeniería del Software y Bases de Datos, JISBD 2016*, 539-550.

Carrizo, D., y Rojas, J. (2018). Metodologías, técnicas y herramientas en ingeniería de requisitos: Un mapeo sistemático. *Ingeniare – Revista Chilena de Ingeniería*, 26(3), 473-485.

Caserman, P., Hoffmann, K., Müller, P., Schaub, M., Straßburg, K., Wiemeyer, J., Bruder, R., y Göbel, S. (2020). Quality criteria for serious games: Serious part, game part, and balance. *JMIR Serious Games*, 8(3), e19037.

Chanchí, G., Gómez, M., y Campo, W. (2019). Propuesta de un videojuego educativo para la enseñanza-aprendizaje de la clasificación de requisitos en ingeniería de software. *Revista Ibérica de Sistemas y Tecnologías de la Información*, E22(8), 1-4.

Chaves, V., y Weiler, C. (2016). Los estudios de casos como enfoque metodológico. *ACADEMO Revista de Investigación en Ciencias Sociales y Humanidades*, 3(2), 1-11.

Chin, C., Kamsani, N., Zhong, X., Cui, R., y Yang, C. (2018, July). Unity3D serious game engine for high fidelity virtual reality training of remotely-operated vehicle pilot. En *2018 10th International Conference on Modelling, Identification and Control (iCMiC)* (pp. 1-6). IEEE.

Chipia, J. (2011). Juegos serios: Alternativa innovadora. *Conocimiento Libre y Educación (CLEDE)*, 2(2), 1-18.

CMMI Product Team (2018). Capability Maturity Model Integration (CMMI) for Development, Version 2.0, Software Engineering Institute, Carnegie Mellon University.

Coventry, T. (2015). Requirements management – Planning for success!: Techniques to get it right when planning requirements. Paper presented at PMI® Global Congress 2015—EMEA, London, England. Newtown Square, PA: Project Management Institute.

Cruz, M., Bernárdez, B., Durán, A., Galindo, J., y Ruiz-Cortés, A. (2019). Replication of studies in empirical software engineering: A systematic mapping study, from 2013 to 2018. *IEEE Access*, 8, 26773-26791.

Dalpiaz, F., y Cooper, K. (2018). Games for requirements engineers: Analysis and directions. *IEEE Software*, 37(1), 50-59.

Daun, M., Grubb, A., Stenkova, V., y Tenbergen, B. (2023). A systematic literature review of requirements engineering education. *Requirements Engineering*, 28(2), 145-175.

de Carvalho, C., y Coelho, A. (2022). Game-based learning, gamification in education and serious games. *Computers*, 11(3), 36.

De la Cruz, C., y Castro, G. (2015). Metodología para la adquisición y gestión de requerimientos en el desarrollo de software para pequeñas y medianas empresas (Pymes) del departamento de Risaralda. Tesis. Magister en Ingeniería de Sistemas y Computación. Facultad de Ingeniería. Universidad Tecnológica de Pereira.

de Lope, R., Medina-Medina, N., Paderewski, P., y Vela, F. (2015). Design methodology for educational games based on interactive screenplays. *Congreso de la Sociedad Española para las Ciencias del Videojuego, CoSECiVi 2015, 1394*, 90-101.

Dick, J., Hull, E. y Dick, J. (2017). *Requirements Engineering (4th Edition)*. Cham, Switzerland: Springer International Publishing.

Durmic, N. (2020). Factors influencing project success: A qualitative research. *TEM Journal, 9(3)*, 1011-1020.

Ebert, C. (2018). 50 years of software engineering: Progress and perils. *IEEE Software, 35(5)*, 94-101.

Epifânio, J., Miranda, É., Trindade, G., Lucena, M., y Silva, L. (2019, September). A qualitative study of teaching requirements engineering in universities. En *Proceedings of the XXXIII Brazilian Symposium on Software Engineering* (pp. 161-165). ACM.

Franch, X., Méndez, D., Vogelsang, A., Heldal, R., Knauss, E., Oriol, M., Travassos, G. H., Carver, J. C., y Zimmermann, T. (2022). How do practitioners perceive the relevance of requirements engineering research? *IEEE Transactions on Software Engineering, 48(6)*, 1947-1964.

Freeman, R. E y McVea, J. (2005). A stakeholder approach to strategic management. En Hitt, M. A., Freeman, R., y Harrison, J. S. (Eds.), *The Blackwell handbook of strategic management*, New York, NY: Wiley & Sons (pp. 183-201).

Gambo, I., Ikono, R., Achimugu, P., y Iroju, O. (2018). A ranking model for software requirements prioritization during requirements engineering: A case study. *International Journal of Computer Science and Information Security (IJCSIS), 16(4)*, 255-268.

Ganesh, L. (2014, December). Board game as a tool to teach software engineering concept--Technical debt. En *2014 IEEE Sixth International Conference on Technology for Education* (pp. 44-47). IEEE.

Ganesh, T. G., y Schnittka, C. G. (2014). Engineering education in the middle grades. En: Purzer, S., Strobel, J., y Cardella, M. (Eds.), *Engineering in pre-college settings: Synthesizing research, policy, and practices*, Purdue University Press (pp. 89-115).

García, I., Pacheco, C., León, A., y Calvo-Manzano, J. (2019). Experiences of using a game for improving learning in software requirements elicitation. *Computer Applications in Engineering Education, 27(1)*, 249-265.

García, I., Pacheco, C., Leon, A., y Calvo-Manzano, J. (2020). A serious game for teaching the fundamentals of ISO/IEC/IEEE 29148 systems and software engineering--Lifecycle processes--Requirements engineering at undergraduate level. *Computer Standards & Interfaces, 67*, 103377.

Garousi, V., Giray, G., Tüzün, E., Catal, C., y Felderer, M. (2019). Aligning software engineering education with industrial needs: A meta-analysis. *Journal of Systems and Software, 156*, 65-83.

Gasca-Hurtado, G., Gómez-Álvarez, M., y Manrique-Losada, B. (2019). Using gamification in software engineering teaching: Study case for software design. En: Rocha, Á., Adeli, H., Reis, L., Costanzo, S. (Eds.), *New Knowledge in Information Systems and Technologies: Volume 3*, Springer International Publishing (pp. 244-255).

Georgieva-Tsaneva, G. (2019). Serious games and innovative technologies in medical education in Bulgaria. *TEM Journal*, 8(4), 1398-1403.

Gibbs, W. (1994). Software's chronic crisis. *Scientific American*, 271(3), 86-95.

Glinz, M., van Loenhoud, H., Staal, S., y Bühne, S. (2020). Handbook for the CPRE Foundation Level according to the IREB Standard – Education and Training for Certified Professional for Requirements Engineering. Version 1.0. International Requirements Engineering Board.

Gómez-Álvarez, M., y Zapata Jaramillo, C. (2022). Una propuesta de clasificación de juegos con propósito educativo para ingeniería de software. *Ingeniare – Revista Chilena de Ingeniería*, 30(2), 239-254.

Gul, S., Dar, H., Malik, A., Zulqarnain, M., y Imtiaz, S. (2021). Gamification and gaming elements for software requirements elicitation: A systematic literature review. *International Journal on Electrical Engineering and Informatics*, 13(4), 931-950.

Henriksen, D., Richardson, C., y Mehta, R. (2017). Design thinking: A creative approach to educational problems of practice. *Thinking skills and Creativity*, 26, 140-153.

Henry, J., Tang, S., Hanneghan, M., y Carter, C. (2018, May). A framework for the integration of serious games and the Internet of Things (IoT). En *2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH)* (pp. 1-8). IEEE.

Herranz, E., Guzmán, J., de Amescua-Seco, A., y Larrucea, X. (2019). Gamification for software process improvement: a practical approach. *IET Software*, 13(2), 112-121.

Hertz, K., y Spoletini, P. (2018, August). Are requirements engineering courses covering what industry needs? A preliminary analysis of the United States situation. En *2018 IEEE 8th International Workshop on Requirements Engineering Education and Training (REET)* (pp. 20-23). IEEE.

Hussain, A., Mkpojiogu, E., y Kamal, F. (2016). The role of requirements in the success or failure of software projects. *International Review of Management and Marketing*, 6(7), 306-311.

Ibrahim, Z., Soo, M., Soo, M., y Aris, H. (2019, December). Design and development of a serious game for the teaching of requirements elicitation and analysis. En *2019 IEEE international conference on engineering, technology and education (TALE)* (pp. 1-8). IEEE.

IEEE Computer Society. (2014). Software Engineering Competency Model (SWECOM), Version 1.0. <https://dahlan.unimal.ac.id/files/ebooks/SWECOM.pdf>

IEEE Computer Society and Association for Computing Machinery. (2015). Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (SE2014). <https://www.acm.org/binaries/content/assets/education/se2014.pdf>

Incencio Piñeiro, G. S., Guerra Cantero, L. M, y Lissabet Rivero, J. L. (2022). Evolución histórica del proceso de enseñanza-aprendizaje de la asignatura Ingeniería de Software I. *Didáctica y Educación*, 13(2), 91-105.

ISO/IEC/IEEE (2018). Systems and Software Engineering — Life Cycle Processes — Requirements Engineering (ISO/IEC/IEEE 29148:2018).

Johnson, J. (2018). Decision Latency Theory: It's All about the Interval. Standish Group International, Incorporated.

Johnson, J. (2020). Chaos report 2020: Beyond Infinity. Standish Group International, Incorporated.

Khade, V., Masoudi, N., Acena, D., Freeman, G., Rai, R., Gorsich, D., Rizzo, D., y Castanier, M. (2022, November). Requirements elicitation: Impacts of gamification on variety, novelty, and

completeness. En *Proceedings of the ASME International Mechanical Engineering Congress & Exposition* (pp. 254-262). American Society of Mechanical Engineers.

Khan, H., Malik, M., Alotaibi, Y., Alsufyani, A., y Alghamdi, S. (2021). Crowdsourced requirements engineering challenges and solutions: A software industry perspective. *Computer Systems Science & Engineering*, 39(2), 221-236.

Kolpondinos, M., y Glinz, M. (2020). GARUSO: A gamification approach for involving stakeholders outside organizational reach in requirements engineering. *Requirements Engineering*, 25(2), 185-212.

Krathwohl, D. (2002). A revision of Bloom's taxonomy: An overview. *Theory into practice*, 41(4), 212-218.

Kurkovsky, S., Ludi, S., y Clark, L. (2019, February). Active learning with LEGO for software requirements. En *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 218-224). ACM.

Larios, V. M, González, O. F., Zavala, M. E., García, C. A., y Martínez, M. P. (2015). Producción de videojuegos serios. Guadalajara, México: Universidad de Guadalajara.

Larson, K. (2020). Serious games and gamification in the corporate training environment: A literature review. *TechTrends*, 64(2), 319-328.

Lima, R., Filippetto, A., Heckler, W., Barbosa, J., y Leithardt, V. (2022). Towards ubiquitous requirements engineering through recommendations based on context histories. *PeerJ Computer Science*, 8, e794.

Liu, Z., Li, B., Wang, J., y Yang, R. (2021). Requirements engineering for crossover services: Issues, challenges and research directions. *IET Software*, 15(1), 107-125.

Londoño Vásquez, L., y Rojas López, M. (2020). From games to gamification: A proposal for an integrated model. *Educación y Educadores*, 23(3), 493-512.

Loucopoulos, P., y Karakostas, V. (1995). *System Requirements Engineering*. Boston, MA: McGraw-Hill, Inc..

Marques, P., Silva, M., Gusmão, C., Castro, D., y Schots, M. (2020, November). Requirements engineering out of the classroom: Anticipating challenges experienced in practice. En *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)* (pp. 1-9). IEEE.

Martins, L., Ossada, J., Belgamo, A., y Ranieri, B. (2013, March). Requirements elicitation guide for embedded systems: An industry challenge. En *8th International Conference on Software Engineering Advances (ICSEA)* (pp. 106-111). IEEE.

Maté, J. L., y Silva, A. (2005). *Requirements engineering for sociotechnical systems*. London, UK: IGI Global.

Mayer, B., y Weinreich, R. (2019, April). The effect of gamification on software architecture knowledge management: A student experiment and focus group study. En *34th ACM/SIGAPP Symposium on Applied Computing* (pp. 1731-1740). ACM.

Méndez, D. (2018). Supporting requirements-engineering research that industry needs: The NaPiRE initiative. *IEEE Software*, 35(01), 112-116.

Méndez, D., Wagner, S., Kalinowski, M., Felderer, M., Mafra, P., Vetrò, A., Conte, T., Christiansson, M. T., Greer, D., Lassenius, C., Männisto, T., Nayabi, M., Oivo, M., Penzenstadler, B., Pfahl, D., Prokladnicki, R., Ruhe, G., Schekelmann, A., Sen, S., Spinola, R., Tuzcu, A., de la Vara, J. L., y Wieringa, R. (2017). Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice. *Empirical software engineering*, 22, 2298-2338.

Mi, Q., Keung, J., Mei, X., Xiao, Y., y Chan, W. (2018, July). A gamification technique for motivating students to learn code readability in software engineering. *2018 International Symposium on Educational Technology (ISET)* (pp. 250-254). IEEE.

Mohabuth, A. Q. (2019, June). Software practitioners challenges in the requirement engineering phase of software development. En *15th International Conference on Advances in Science, Engineering & Natural Resources (PSENR-19)* (pp. 45-48). IFEP.

Monsalve, E., Werneck, V., y Leite, J. C. S. D. P. (2010). Evolución de un juego educacional de ingeniería de software a través de técnicas de elicitación de requisitos. En *Proceedings of XIII Workshop on Requirements Engineering (WER'2010)* (pp. 12-23). UC.

Monsalve, E., Toro, M., Mazo, R., Velasquez, D., Vallejo, P., Cardona, J. F., Rincón, R., Werneck, V. M., y Leite, J. C. S. D. P. (2018). SimulES-W: A collaborative game to improve software engineering teaching. *Computación y Sistemas*, 22(3), 953-983.

Moody, D., y Sindre, G. (2003, June). Evaluating the effectiveness of learning interventions: An information systems case study. En *11th European Conference on Information Systems. Association for Information Systems (ECIS)* (pp. 235-242). AIS.

Morales-Trujillo, M., y García-Mireles, G. (2020). Gamification and SQL: An empirical study on student performance in a database course. *ACM Transactions on Computing Education (TOCE)*, 21(1), 1-29.

Moyano, E., Jerez, M., Romano, L., Villarreal, M., Moncho, M., Villanueva, T., Rigoni, B., Andrade, D., y Iost, D. (2021). Diseño de estrategias didácticas para la enseñanza de ingeniería de requerimientos. *XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021, Chilecito, La Rioja)* (pp. 324-328). Red de Universidades con Carreras en Informática.

Naik, N., y Jenkins, P. (2019, August). Relax, it's a game: Utilising gamification in learning agile scrum software development. In *2019 IEEE Conference on Games (CoG)* (pp. 1-4). IEEE.

Nowostawski, M., McCallum, S., y Mishra, D. (2018). Gamifying research in software engineering. *Computer Applications in Engineering Education*, 26(5), 1641-1652.

Nwokeji, J., Aqlan, F., Martinez, J., Holmes, T., Frezza, S., y Orji, R. (2018, October). Panel: Integrating Requirements Engineering Education into Core Engineering Disciplines. In *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1-3). IEEE.

NYCE (2016) NMX-I-059-2-NYCE-2016 Tecnologías de la información–software–modelos de procesos y evaluación para desarrollo y mantenimiento de software–parte 2: requisitos de procesos (MOPROSOFT) (CANCELA ALA NMX-I-059/02-NYCE-2011).

Ochodek, M., y Kopczyńska, S. (2018). Perceived importance of agile requirements engineering practices—a survey. *Journal of Systems and Software*, 143, 29-43.

Oliveros, A., y Martínez, S. J. (2015). La noción de stakeholder en la ingeniería de requerimientos. *XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021, Chilecito, La Rioja)* (pp. 123-128). Red de Universidades con Carreras en Informática.

Ošlejšek, R., Rusňák, V., Burská, K., Švábenský, V., y Vykopal, J. (2019, October). Visual feedback for players of multi-level capture the flag games: Field usability study. En *2019 IEEE Symposium on Visualization for Cyber Security (VizSec)* (pp. 1-11). IEEE.

Pacheco, C., y García, I. (2012). A systematic literature review of stakeholder identification methods in requirements elicitation. *Journal of Systems and Software*, 85(9), 2171-2181.

Pacheco, C., y Tovar, E. (2007). Stakeholder identification as an issue in the improvement of software requirements quality. En: Krogstie, J., Opdahl, A., Sindre, G. (Eds.), *Advanced Information Systems Engineering*. Springer Berlin Heidelberg (pp. 370-380).

Parizi, R., y Dehghantanha, A. (2018, August). On the understanding of gamification in blockchain systems. En *6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)* (pp. 214-219). IEEE.

Perini, A., Seyff, N., Stade, M., y Susi, A. (2018, August). Exploring RE knowledge for gamification: Can RE achieve a high score? *1st International Workshop on Affective Computing for Requirements Engineering (AffectRE)* (pp. 14-19). IEEE.

Petri, G., von Wangenheim, C., y Borgatto, A. (2021). Evolution of a model for the evaluation of games for software engineering education. En: Cooper, K. (Ed.), *Software Engineering Perspectives in Computer Game Development*. Chapman and Hall/CRC (pp. 103-137).

Piattini, M. (2016). Evolución de la ingeniería del software y la formación de profesionales. *Bit & Byte*, 2(4), 15-17.

Pinelle, D., Wong, N., y Stach, T. (2008, April). Heuristic evaluation for games: Usability principles for video game design. En *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)* (pp. 1453-1462). ACM.

Piras, L. (2018). Agon: a gamification-based framework for acceptance requirements. Tesis de Doctorado. Departamento de Ingeniería de la Información y Ciencias Computacionales. Universidad Degli Studi Di Trento.

Pressman, R., y Maxim, B. (2019). *Software Engineering – A practitioner’s Approach*. Boston, MA: McGraw-Hill.

Rasheed, A., Zafar, B., Shehryar, T., Aiman, N., Sajid, M., Ali, N., Hanif, S., y Khalid, S. (2021). Requirement engineering challenges in agile software development. *Mathematical Problems in Engineering*, 2021, 1-18.

Robertson, S., y Robertson, J. (2012). *Mastering the requirements process: Getting requirements right*. Upper Saddle River, NJ: Addison-Wesley.

Rodrigues, P., Souza, M., y Figueiredo, E. (2018, October). Games and gamification in software engineering education: A survey with educators. En *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE.

Romero, M., Vizcaíno, A., y Piattini, M. (2008). Un simulador para la enseñanza de la elicitación de requisitos en el contexto del desarrollo global del software. XIII Jornadas de Ingeniería del Software y Bases de Datos (*JISBD, Gijón, España*) (pp. 397-402). Asociación de Técnicos en Informática.

Serna, E., y Serna, A. (2019). Madurez de la gestión del conocimiento en la ingeniería de requisitos. *Revista Ibérica de Sistemas y Tecnologías de la Información*, (E17), 123-141.

Sevilla, G., Zapata, S., Torres, E., y Giraldo, F. (2018). Propuesta de modelo de proceso de ingeniería distribuida de requisitos de software. *Revista Tecnología y Ciencia*, (33), 175-193.

Sevilla, G., Zapata, S., Torres, E., Fáber, G., y Gallardo, F. (2019). Un marco para la enseñanza de ingeniería distribuida de requisitos en las universidades. *XXI Workshop de Investigadores en Ciencias de la Computación (WICC 2019, Universidad Nacional de San Juan)* (pp. 125-130). Red de Universidades con Carreras en Informática.

Shafiq, M., Zhang, Q., Akbar, M., Khan, A. A., Hussain, S., Amin, F., Khan, A., y Soofi, A. A. (2018). Effect of project management in requirements engineering and requirements change management processes for global software development. *IEEE Access*, 6, 25747-25763.

Sommerville, I. (2019). *Engineering software products: An introduction to modern software engineering*. Boston, MA: Pearson.

Trickel, E., Disperati, F., Gustafson, E., Kalantari, F., Mabey, M., Tiwari, N., Safaei, Y., Doupe, A., y Vigna, G. (2017, August). Shell we play a game? {CTF-as-a-service} for security education. En *2017 USENIX Workshop on Advances in Security Education (ASE 17)* (pp. 1-10) NSF.

Tripathi, N., Klotins, E., Prikladnicki, R., Oivo, M., Pompermaier, L. B., Kudakacheril, A. S., Unterkalmsteiner, M., Liukkunen, K., y Gorschek, T. (2018). An anatomy of requirements engineering in software startups using multi-vocal literature and case survey. *Journal of Systems and Software, 146*, 130-151.

Unity (2016). *Manual de Unity. Version 5.3 (switch to 5.4b)*. Unity documentation. <https://docs.unity3d.com/es/530/Manual/UnityManual.html>.

Vera, R. (Editor) (2019). *Ingeniería de software en México: Educación, industria e investigación*. México: Academia Mexicana de Computación.

Visvizi, A., y Lytras, M. (Editors). (2019). *Research & innovation forum 2019: Technology, innovation, education, and their social impact*. Cham, Switzerland: Springer Nature.

Vizcaíno, A., García, F., Guzmán, I., y Moraga, M. (2019). Evaluating GSD-aware: A serious game for discovering global software development challenges. *ACM Transactions on Computing Education (TOCE), 19(2)*, 1-23.

Vogelsang, A., y Borg, M. (2019, September). Requirements engineering for machine learning: Perspectives from data scientists. En *IEEE 27th International Requirements Engineering Conference Workshops (REW)* (pp. 245-251). IEEE.

Weyer, T., Daun, M., y Tenbergen, B. (2020). The changing world and the adapting machine: How digital transformation changes requirements engineering in the embedded and cyberphysical systems industry. *IEEE Software, 38(5)*, 83-91.

Wieggers, K., y Beatty, J. (2013). *Software requirements*. London, UK: Pearson Education.

Wohlin, C. (2021). Case study research in software engineering—It is a case, and it is a study, but is it a case study? *Information and Software Technology, 133*, 106514.

Wohlin, C., y Rainer, A. (2022). Is it a case study?—A critical analysis and guidance. *Journal of Systems and Software, 192*, 111395.

Wong, L. R. (2019). Contribuciones en el proceso de elicitación de requisitos: Factores, actividades y cualidades. Tesis. Doctorado en Ingeniería de Sistemas e Informática. Facultad de Ingeniería de Sistemas e Informática. Universidad Nacional Mayor de San Marcos.

Zahid, A., Liaqat, A., Farooq, M., y Naseer, S. (2020). Requirement elicitation issues and challenges in Pakistan software industry. *VFAST Transactions on Software Engineering 8(1)*, 28-36.

Zainuddin, F., Arshah, R., y Mohamad, R. (2019, January). Software visual specification for requirement engineering education. En *10th International Conference on E-Education, E-Business, E-Management and E-Learning (IC4E)* (pp. 235-240). ACM.

Zhonggen, Y. (2019). A meta-analysis of use of serious games in education over a decade. *International Journal of Computer Games Technology, 2019*, 1-8.

7. Anexo A.- Acrónimos

ACM	Asociación de Maquinaria Computacional
ADDIE	Análisis, Diseño, Desarrollo, Implementación y Evaluación
AHP	Proceso de Jerarquía Analítica
AK	Conocimiento sobre Arquitectura de Software
ATM	Modo de Transferencia Asíncrona
CC2020	Plan de Estudios de Computación 2020
CMMI-DEV	Modelo de Capacidad y Madurez Integrado para el Desarrollo
CPU	Unidad Central de Procesamiento
CTF	Capturar una Bandera
DA	De Acuerdo
DTA	Pensamiento de Diseño y Agon
ED	En Desacuerdo
EEUU	Estados Unidos
ER	Elicitación de Requisitos
FSE	Fundamentos de Ingeniería de Software
GBL	Aprendizaje Basado en Juegos
GPA	Promedios de las Calificaciones Acumuladas
GSD	Desarrollo Global de Software
IE	Inteligencia Emocional
IEC	Comisión Electrotécnica Internacional
IEEE	Instituto de Ingenieros Eléctricos y Electrónicos
IoT	Internet de las Cosas
IR	Ingeniería de Requisitos
IS	Ingeniería de Software
ISO	Organización Internacional de Normalización
JS	Juegos Serios
JSI	Juegos Serios Inteligentes
M	Media

MEEGA	Modelo de Evaluación de Juegos Educativos
MSLQ	Cuestionario de Estrategias Motivadas para el Aprendizaje
N	Neutral
NaPiRE	Nombrar el Dolor en la Ingeniería de Requisitos
NPC	Personaje No Jugable
NYCE	Normalización y Certificación
OTAN	Organización del Tratado del Atlántico Norte
PACAS	Gestión Participativa del Cambio de Arquitectura de Sistemas ATM
REE	Educación en la Ingeniería de Requisitos
ROV	Vehículos Operados a Distancia
RSL	Revisión Sistemática de Literatura
SAD	Análisis y Diseño de Sistemas
SAS	Sistemas Auto Adaptativos
SD	Desviación Estándar
SEA	Solicitud de Participación Estudiantil
SE2014	Lineamientos Curriculares para Programas de Licenciatura en IS
SGOMM	Metodología de Modelado Semi-formal Orientada a Objetivos
SPI	Mejora de Procesos de Software
SQL	Lenguaje de Consulta Estructurada
SWEBOK	Cuerpo de Conocimiento de la Ingeniería de Software
SWECOM	Modelo de Competencias de la IS
TA	Totalmente de Acuerdo
TD	Totalmente en Desacuerdo
UCD	Diseño Centrado en el Usuario
UCM	Metamodelo de Contexto de Usuario
UML	Lenguaje Unificado de Modelado
UTM	Universidad Tecnológica de la Mixteca