



## **UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA**

**“DISEÑO, DESARROLLO, Y EVALUACIÓN DE UN JUEGO SERIO  
PARA SOPORTAR LA ENSEÑANZA/APRENDIZAJE DEL  
ESTÁNDAR ISO/IEC/IEEE 26515:2018 *SYSTEMS AND SOFTWARE  
ENGINEERING — DEVELOPING INFORMATION FOR USERS IN AN  
AGILE ENVIRONMENT* EN UN CONTEXTO UNIVERSITARIO”**

### **TESIS**

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

### **PRESENTA**

**MARCO ANTONIO HERNÁNDEZ SANTOS**

### **DIRECTOR DE TESIS**

**DR. IVÁN ANTONIO GARCÍA PACHECO**

### **CODIRECTORA DE TESIS**

**DRA. CARLA LENINCA PACHECO AGÜERO**

**HUAJUAPAN DE LEÓN, OAX. SEPTIEMBRE DE 2024**



**Tesis presentada el 30 de septiembre de 2024**

**ante los siguientes sinodales:**

M.M.I. Gerardo Cruz González

M.C. Ricardo Ruiz Rodríguez

M. I. S. Hugo Enrique Martínez Cortés

**Director de tesis:**

Dr. Iván Antonio García Pacheco

**Codirectora de tesis:**

Dra. Carla Leninca Pacheco Agüero



## **Dedicatoria**

Con cariño a mis padres, por ser el pilar de mi vida. Papá, por tu apoyo constante y por enseñarme la perseverancia con tu ejemplo. Mamá, por tu amor incondicional, esfuerzo y sacrificio, esenciales para alcanzar este sueño. Ustedes han sido mi fortaleza y la luz que ha guiado mi camino; este triunfo es tanto mío como suyo.



## **Agradecimientos**

Por medio de estas líneas quiero expresar mi más profundo agradecimiento a las siguientes personas:

A la Universidad Tecnológica de la Mixteca, por brindarme la oportunidad de formar parte de su comunidad académica y por ofrecerme un entorno de aprendizaje enriquecedor.

Al Dr. Iván Antonio García Pacheco, mi director de tesis, por su incansable dedicación y por el valioso tiempo que me brindó a lo largo de este proceso. Agradezco todo el apoyo que me ha brindado, su disposición para orientarme y por compartir conmigo su amplio conocimiento.

A la Dra. Carla Lenínca Pacheco Agüero, mi codirectora de tesis, por su guía constante y sus valiosos consejos. Su dedicación y atención a cada detalle del proyecto han sido esenciales para alcanzar este logro.

Al M.E.C. Josué Neftalí García Matías, por el tiempo dedicado a la revisión de estilo y formato del documento de tesis y por sus contribuciones para que éste fuera el correcto.

A los miembros del comité revisor y evaluador de este trabajo de tesis, por el tiempo y el esfuerzo invertidos en la revisión y evaluación de mi investigación.

A todos los profesores y compañeros que conocí a lo largo de mi carrera, por el valor de sus conocimientos y experiencias, y por la ayuda y la amistad que me brindaron en cada etapa.

A mis queridos padres, Elizabeth Santos Aquino y Juan Manuel Hernández Cruz, por su amor incondicional y constante aliento durante mi formación académica. Su confianza en mí y su constante motivación han sido la base sobre la cual he construido este logro.

A mi amada Sonia Rybicka, por tu amor y dedicación a lo largo de este proceso. Tu confianza en mí, tu paciencia y tu cariño han sido una fuente de gran fortaleza para superar los desafíos y seguir adelante.



## Índice

Índice .....	xi
Lista de tablas .....	xv
Lista de figuras .....	xvii
Resumen .....	xix
1. Introducción.....	1
1.1. Contexto del problema.....	1
1.2. Importancia del problema.....	2
1.3. Necesidad de la solución .....	4
1.4. Delimitaciones de la tesis .....	6
1.5. Hipótesis del trabajo .....	7
1.6. Objetivos del trabajo.....	7
1.6.1. Objetivo general .....	7
1.6.2. Objetivos específicos.....	7
1.7. Metodología.....	7
1.7.1. Diseño e implementación de un juego serio.....	8
1.7.2. Realización de una evaluación empírica .....	10
1.7.2.1. Participantes .....	10
1.7.2.2. Instrumentos .....	10
1.7.2.3. Proceso de evaluación .....	11
1.8. Estructura de la tesis.....	12
2. Marco teórico.....	13
2.1. Uso de estándares en la industria de software .....	13
2.2. Enseñanza de los estándares en un contexto universitario .....	16
2.3. El estándar ISO/IEC/IEEE 26515:2018 <i>Systems and software engineering — Developing information for users in an agile environment</i> .....	18
2.3.1. Historia del estándar .....	19
2.3.2. Descripción del estándar.....	20
2.3.3. Estructura del estándar.....	21
2.4. Estado del arte .....	24

2.4.1. Enseñando el estándar ISO/IEC 12207:1995 <i>Systems and software engineering – Software lifecycle processes</i> : Un enfoque basado en un juego serio .....	25
2.4.1.1. Objetivo .....	25
2.4.1.2. Descripción .....	25
2.4.1.3. Resultados.....	27
2.4.2. Un juego serio para apoyar la enseñanza del estándar ISO 21500:2012 en el contexto de la gestión de proyectos de software .....	28
2.4.2.1. Objetivo .....	28
2.4.2.2. Descripción .....	28
2.4.2.3. Resultados.....	30
2.4.3. ProcSoft: Un juego de tablero para enseñar los procesos de software en base al estándar ISO/IEC 29110 .....	32
2.4.3.1. Objetivo .....	32
2.4.3.2. Descripción .....	32
2.4.3.3. Resultados.....	34
2.4.4. Un juego serio para enseñar los conceptos fundamentales de la Ingeniería de Requisitos y del estándar ISO/IEC/IEEE 29148:2011 a nivel licenciatura.....	35
2.4.4.1. Objetivo .....	35
2.4.4.2. Descripción .....	35
2.4.4.3. Resultados.....	38
2.4.5. Estrategia basada en un juego serio para la capacitación a nivel licenciatura en el estándar ISO/IEC 29110:2018 .....	40
2.4.5.1. Objetivo .....	40
2.4.5.2. Descripción .....	40
2.4.5.3. Resultados.....	42
2.5. Consideraciones finales sobre el estado del arte.....	44
3. Diseño y desarrollo del juego serio .....	45
3.1. Aspectos metodológicos .....	45
3.1.1. Pre-1: Diseño inicial de los objetivos educativos y las competencias básicas .....	47
3.1.1.1. Recolección de opiniones y profesores sobre los métodos de enseñanza .....	48
3.1.1.2. Análisis de planes de estudio y temarios para identificar los objetivos educativos iniciales .....	55
3.1.1.3. Identificación de las competencias básicas.....	60
3.1.2. Pre-2: Diseño del tipo de juego .....	65
3.1.2.1. Género y plataforma del juego .....	65
3.1.2.2. Propósito y público objetivo.....	66
3.1.2.3. Área de aplicación .....	66
3.1.3. Pre-3: Diseño inicial de la historia y los personajes .....	66
3.1.3.1. Flujo del juego .....	67
3.1.3.2. Personajes .....	71
3.1.4. Fase 1: Diseño de los capítulos.....	71

---

3.1.5. Fase 2: Diseño de la escena .....	73
3.1.5.1. Level_0_MainMenu .....	74
3.1.5.1.1. Diseño del escenario.....	74
3.1.5.1.2. Personajes .....	74
3.1.5.2. Level_1 .....	74
3.1.5.2.1. Diseño del escenario.....	75
3.1.5.2.2. Personajes .....	75
3.1.5.2.3. Diálogos y retos del juego .....	75
3.1.5.3. Level_2 .....	76
3.1.5.3.1. Diseño del escenario.....	76
3.1.5.3.2. Personajes .....	77
3.1.5.3.3. Diálogos y retos del juego .....	77
3.1.5.4. Level_3 .....	77
3.1.5.4.1. Diseño del escenario.....	77
3.1.5.4.2. Personajes .....	78
3.1.5.4.3. Diálogos y retos del juego .....	78
3.1.6. Fase 3: Identificación de los retos educativos y su evaluación .....	79
3.1.7. Fase 4: Identificación/clasificación de emociones .....	81
3.1.7.1. Emociones presentes en la escena Level_0_MainMenu .....	81
3.1.7.2. Emociones presentes en las escenas Level_1, Level_2 y Level_3.....	82
3.1.8. Fase 5: Diseño de la adaptación .....	83
3.1.9. Fase 6: Diseño de la colaboración .....	83
3.1.9.1. Implementación del juego .....	83
3.1.9.2. Herramientas y lenguaje de programación .....	83
3.1.9.3. Desarrollo .....	84
3.2. Comentarios finales sobre el desarrollo del juego serio .....	84
4. Resultados de la evaluación empírica.....	87
4.1. Participantes .....	87
4.2. Instrumentos .....	88
4.3. Estrategia de evaluación .....	92
4.4. Resultados de la evaluación.....	93
4.5. Conclusiones sobre los resultados de la evaluación .....	107
5. Conclusiones .....	109
5.1. Conclusiones derivadas de la investigación .....	109
5.2. Conclusiones derivadas de la evaluación empírica .....	111
5.3. Aspectos para mejorar en el juego serio propuesto .....	112
6. Bibliografía.....	115
7. Anexo A.- Acrónimos .....	121
8. Anexo B.- Preguntas incluidas en el juego serio.....	123

8.1. Estructura de las preguntas ..... 123

## Lista de tablas

Tabla 1. Procesos y actividades establecidas por el estándar ISO/IEC/IEEE 26515 (2018) .....	22
Tabla 2. Conteo de puntos finales del proceso creado (traducido de Moura y Santos, 2018) .....	33
Tabla 3. Cuestionario para opiniones de los profesores sobre los métodos de enseñanza (adaptada de León, 2016).....	48
Tabla 4. Cuestionario para opiniones de los estudiantes sobre los métodos de enseñanza.....	52
Tabla 5. Análisis sobre el contenido de los temarios y lineamientos para la carrera de Ingeniería en Computación.....	57
Tabla 6. Análisis sobre el contenido de los temarios y lineamientos para la carrera de Licenciatura en Informática.....	57
Tabla 7. Tópicos relacionados con el área de conocimiento “Proceso de software” (adaptada de ACM-IEEE, 2014).....	60
Tabla 8. Lineamientos y recomendaciones del SWECOM para la enseñanza de los métodos ágiles (resumida de IEEE-SC, 2014) .....	62
Tabla 9. Actividades y niveles de competencia para desarrollar habilidades con los métodos ágiles (resumida de IEEE-SC, 2014) .....	63
Tabla 10. Retos educativos, mecánicas y dinámicas de juego .....	80
Tabla 11. Cuestionario para determinar el nivel de comprensión de los estudiantes .....	88
Tabla 12. Instrumento de medición de MEEGA+ (traducido y adaptado de Petri et al., (2019))... 90	
Tabla 13. Cuestionario para recoger las percepciones de los profesores (adaptado de Bourgonjon et al., (2013) y Bodnar et al., (2015)).....	92
Tabla 14. Percepciones de los estudiantes sobre la dimensión de usabilidad con MEEGA+ .....	97
Tabla 15. Percepciones de los estudiantes sobre la dimensión de confianza con MEEGA+ .....	99
Tabla 16. Percepciones de los estudiantes sobre la dimensión de desafío con MEEGA+ .....	99
Tabla 17. Percepciones de los estudiantes sobre la dimensión de satisfacción con MEEGA+.....	100
Tabla 18. Percepciones de los estudiantes sobre la dimensión de diversión con MEEGA+.....	101
Tabla 19. Percepciones de los estudiantes sobre la dimensión de atención enfocada con MEEGA+ .....	102
Tabla 20. Percepciones de los estudiantes sobre la dimensión de relevancia con MEEGA+ .....	103
Tabla 21. Percepciones de los estudiantes sobre la dimensión de aprendizaje percibido con MEEGA+.....	104
Tabla 22. Percepciones de los profesores sobre el juego propuesto.....	106



## Lista de figuras

Figura 1.1. Metodología de desarrollo de juegos basada en guiones interactivos (de Lope et al., (2015), como se citó en (León, 2016)) .....	9
Figura 2.1. Diálogo entre jugador y NPC responsable del piso relacionado con los procesos de verificación y validación de software (tomada de Aydan et al., 2017) .....	26
Figura 2.2. Diagrama de procesos seleccionados del estándar ISO/IEC 12207:1995 y su secuencia de ocurrencia durante el juego (traducida de Aydan et al., 2017) .....	27
Figura 2.3. Representación del ciclo de vida de “ProDec” (traducida de Calderón et al., 2018a)..	29
Figura 2.4. Pantalla de ejemplo de la etapa de ejecución de “ProDec” (tomada de Calderón et al., 2018a) .....	31
Figura 2.5. Tablero y casillas especiales del juego “ProcSoft” (tomada de Moura y Santos, 2018) .....	33
Figura 2.6. Pantalla inicial de “Requengin” (tomada de García et al., 2020).....	36
Figura 2.7. La estructura de los niveles de “Requengin” (traducida de García et al., 2020).....	37
Figura 2.8. Flujo de juego de “Requengin” (traducida de García et al., 2020) .....	38
Figura 2.9. Pantalla del tablero de “Gestión del Proyecto” (tomada de Bonilla-Rivas et al., 2021)	41
Figura 2.10. Resultados de la evaluación empírica (tomada de Bonilla-Rivas et al., 2021) .....	43
Figura 3.1. Metodología para desarrollar el juego serio (de Lope et al., (2015), como se citó en (León, 2016)) .....	45
Figura 3.2. Opiniones de los profesores con relación a las metodologías tradicionales de enseñanza .....	50
Figura 3.3. Distribución de respuestas en el cuestionario aplicado a los profesores participantes .	51
Figura 3.4. Opiniones de los profesores con relación a los juegos serios como recurso educativo	52
Figura 3.5. Opiniones de los estudiantes con relación a las metodologías tradicionales de enseñanza .....	53
Figura 3.6. Distribución de respuestas en el cuestionario aplicado a los estudiantes participantes	54
Figura 3.7. Opiniones de los estudiantes con relación a los juegos serios como recurso educativo	55
Figura 3.8. Aspectos comunes entre ISO/IEC/IEEE 26515:2018, SE 2014 y SWECOM.....	64
Figura 3.9. Personajes principales del juego .....	67
Figura 3.10. Pantalla inicial del juego .....	67
Figura 3.11. Selección del nivel de dificultad del juego .....	68
Figura 3.12. Pantalla inicial del juego .....	69

---

Figura 3.13. Selección de temas relacionados con las partes del estándar ISO/IEC/IEEE 26515:2018 .....	70
Figura 3.14. Ejemplo de competencia en desarrollo entre tres jugadores .....	70
Figura 3.15. Pantalla inicial del juego .....	71
Figura 3.16. Ejemplo de competencia en desarrollo entre tres jugadores .....	72
Figura 3.17. Flujo de las escenas correspondientes al capítulo diseñado en el juego .....	73
Figura 4.1. Estudiantes tomando las clases teóricas como parte de la evaluación empírica del juego .....	93
Figura 4.2. Resultados de la evaluación sobre el grupo de control mediante un examen de conocimientos .....	94
Figura 4.3. Estudiantes del grupo experimental relacionándose con el juego .....	95
Figura 4.4. Resultados de la evaluación sobre el grupo experimental mediante un examen de conocimientos .....	96
Figura 4.5. Evaluación de la dimensión de usabilidad en el juego propuesto .....	98
Figura 4.6. Evaluación de la dimensión de confianza en el juego propuesto .....	99
Figura 4.7. Evaluación de la dimensión de desafío en el juego propuesto .....	100
Figura 4.8. Evaluación de la dimensión de satisfacción en el juego propuesto.....	101
Figura 4.9. Evaluación de la dimensión de diversión en el juego propuesto.....	102
Figura 4.10. Evaluación de la dimensión de atención enfocada en el juego propuesto .....	103
Figura 4.11. Evaluación de la dimensión de relevancia en el juego propuesto .....	104
Figura 4.12. Evaluación de la dimensión de aprendizaje percibido en el juego propuesto .....	105
Figura 4.13. Percepciones de lo(a)s profesores sobre el juego propuesto .....	105
Figura 8.1. Secciones del estándar y cantidad de preguntas por nivel de complejidad.....	124

## Resumen

Con la llegada de la Industria 4.0 y el uso activo de los métodos ágiles para el desarrollo de software, se ha evidenciado la escasez de técnicas y herramientas apropiadas para la enseñanza de estándares del proceso de software en entornos universitarios. Especialmente aquellos estándares que pretenden concientizar al estudiante de que este tipo de métodos requieren la creación de información que realmente le sea de utilidad al usuario. En este sentido, la mayoría de los métodos tradicionales de enseñanza no abordan situaciones de la vida real y se enfocan principalmente a presentar mucha teoría sin darle importancia al aprendizaje activo, lo que genera a menudo que los estudiantes no se sientan motivados y tengan un bajo desempeño en el curso. En este contexto, con esta tesis se buscó desarrollar un juego serio para fortalecer la enseñanza del estándar *ISO/IEC/IEEE 26515:2018 Systems and software engineering — Developing information for users in an agile environment*, estándar que especifica la forma en la que se puede preparar la información de los usuarios en proyectos que son desarrollados en un entorno ágil. Es importante mencionar que, hasta el día de hoy, no existen propuestas similares, por lo que se utilizó una metodología basada en guiones interactivos para crear un juego serio que sirviera de apoyo educativo a estudiantes universitarios. Finalmente, se diseñó un caso de estudio para evaluar tres variables con participantes reales: motivación, satisfacción, y aprendizaje. Los resultados alcanzados son mostrados para evidenciar la importancia de los juegos serios en la educación de la Ingeniería de Software a nivel universitario.



# 1. Introducción

En este capítulo se abordan aspectos generales que están relacionados con el tema de investigación propuesto en esta tesis. Estos aspectos generales exponen la diferencia existente de conocimiento entre lo que aprende un estudiante en las aulas universitarias y lo requerido para integrarse en la industria, para después explicar la importancia del problema y cuál es la necesidad de darle solución. Además, se definen los objetivos y la hipótesis que guiarán el desarrollo de esta tesis. Como última instancia se presenta de forma general la metodología que se utiliza para realizar esta investigación.

## 1.1. Contexto del problema

En la actualidad existe una diferencia importante entre lo que un estudiante de Ingeniería de Software aprende en las aulas de clases y los conocimientos que requiere para integrarse y trabajar profesionalmente en la industria real. Los estudiantes universitarios recién egresados que logran ingresar al mundo laboral carecen, la mayoría de las ocasiones, de habilidades duras y suaves<sup>1</sup> para adaptarse rápidamente a la forma en que las empresas reales desarrollan el software, en especial las de gran tamaño (Oguz y Oguz, 2019). Estas habilidades no se centran solamente en la programación, también incluyen la capacidad de comunicación, el dominio de las distintas herramientas de software, y la correcta comprensión de los conceptos que se puedan utilizar durante todo el proceso de desarrollo de software. Por lo que, de acuerdo con Cico et al. (2021), tales carencias generan que a los estudiantes recién egresados se les dificulte encontrar un buen trabajo debido a la brecha existente entre las habilidades que poseen y las expectativas de las empresas.

En este sentido, Álvaro Galvis (1992) ya advertía un problema importante hace casi tres décadas al señalar que aprender por uno mismo no era algo que se diera de forma natural ni que fuera consecuencia de solamente asistir a entornos físicos de enseñanza/aprendizaje, ya que es necesario el correcto entendimiento y la aplicación de teorías de aprendizaje humano que den una base para el diseño de ambientes de aprendizaje efectivos. Sin embargo, en la actualidad se puede observar que aún existe una falta de métodos y herramientas que permitan enseñar de manera más real y práctica

---

<sup>1</sup> De acuerdo con Maturro et al. (2019), en el contexto de la Ingeniería de Software, las habilidades técnicas o duras se refieren a aquellas habilidades que poseen las personas para desempeñar funciones especializadas. Es decir, se trata de conocimiento relevante para realizar tareas a menudo complejas, incluyendo las habilidades requeridas para la coordinación del trabajo y el establecimiento de objetivos (e.g., programación, elicitar requisitos, realizar pruebas estructurales al código). Por otro lado, las denominadas habilidades suaves se consideran habilidades interpersonales que permiten que una persona ajuste sus comportamientos en función de las demandas situacionales que influyen en la dinámica de las respuestas a los demás (e.g., liderazgo, comunicación, trabajo en equipo).

cómo funciona un proceso de desarrollo de software. Calderón y Ruiz (2016), por ejemplo, señalaron que los futuros ingenieros de software no sólo debían enfocarse en el conocimiento teórico, también deberían dar la misma importancia a la práctica relacionada con los estándares del proceso de software para concebir, desarrollar, implementar, y mantener un sistema, producto, o servicio de software.

Por si fuera poco, desde el punto de vista de Nakagawa et al. (2021), la cuarta revolución industrial, también llamada Industria 4.0, ha hecho un cambio significativo en los procesos de fabricación puesto que las empresas se enfrentan a la necesidad de desarrollar sistemas más rápidos y continuos, además de buscar una total digitalización, inteligencia, y producción dinámica, logrando una integración de diversas tecnologías como internet de las cosas, computación en la nube, datos masivos, y análisis de datos. Afortunadamente, la presencia de estas tecnologías ofrece las oportunidades para desarrollar y mejorar los entornos de aprendizaje unificando las Tecnologías de la Información (TI) con el proceso educativo, logrando que la brecha existente entre los estudiantes universitarios recién egresados y el mundo exterior se acorte (Barrientos-Avendaño y Areniz-Arévalo, 2019).

Por otro lado, desde la década de los 50 la medición de la calidad del software ha llamado la atención de los especialistas, puesto que anteriormente se prestaba mucha atención a la programación, ignorando las necesidades reales de los usuarios. Durante los años siguientes esta mala práctica se redujo conduciendo al surgimiento de los métodos ágiles para el desarrollo de software, las cuales tienen como objetivo principal proponer un grupo de técnicas tradicionales, modernas, y ágiles para la modelación y creación de software de calidad mediante el uso de heurísticas de construcción y criterios para la comparación de modelos (Montero et al., 2018). Así, con la introducción de los métodos ágiles se dio un giro radical a la situación que prevalecía en 1950, puesto que éstas permitieron una mayor flexibilidad en la industria, se privilegió la comunicación constante con el usuario durante todo el desarrollo, y se propició una mayor y mejor adaptabilidad a los cambios que surgían, logrando así entregas, revisiones, y realimentaciones constantes.

Sin embargo, como se mencionó anteriormente, con la llegada de la Industria 4.0 y el avance de las nuevas tecnologías ha surgido la necesidad apremiante de adquirir conocimiento sólido en los enfoques ágiles de desarrollo de software. De acuerdo con una encuesta realizada por *Forbes Insights Team* y *Scrum Alliance* (2018), el 81% de los líderes ejecutivos consideran que la característica más importante para que una organización sea exitosa es la agilidad, además el 71% de las organizaciones desarrolladoras de software afirman utilizar enfoques ágiles a menudo o siempre. Los resultados de esta misma encuesta evidenciaron que también era necesario que las universidades comenzaran a incluir en sus planes de estudio temas relacionados con el uso de los métodos ágiles, con el objetivo de acercar al estudiante universitario a la industria moderna de software. En este sentido, y como señalaron Kuz et al. (2018), es crucial que las universidades se replanteen la forma tradicional de enseñanza e incorporen espacios de aprendizaje fuera del aula en los cuales los estudiantes puedan adquirir conocimientos sólidos sobre un enfoque ágil a través de proyectos reales de desarrollo de software, con el objetivo de estar mejor preparados para entrar a la competencia laboral del siglo XXI.

## **1.2. Importancia del problema**

Teniendo este objetivo en mente, durante los últimos años se ha generado un número importante de propuestas para mejorar el aprendizaje de los enfoques ágiles de desarrollo a nivel universitario. Por ejemplo, Onieva (2018) mostró cómo el uso de una metodología ágil, como lo es Scrum, aplicada como una herramienta formativa en las escuelas para la realización de proyectos escolares, puede servir como estrategia para el aprendizaje didáctico y como un método para enriquecer las habilidades de trabajo en equipo. Como resultado se generaron entornos de trabajo colaborativos y grupales que

motivaron a los estudiantes a realizar propuestas más innovadoras puesto que se vieron involucrados en el desarrollo de sus propios conocimientos y, al mismo tiempo, se fomentó su capacidad de pensamiento crítico y sus habilidades de comunicación.

Gómez (2020), por otro lado, introdujo una propuesta basada en talleres de producción de software utilizando equipos y métodos ágiles como Scrum y Kanban, con el objetivo de promover una participación e involucramiento más activos de los estudiantes en la resolución de problemas planteados en las aulas de clases, pero que corresponden a empresas reales. Con este estudio se pudo concluir que, al involucrar activamente a los estudiantes en el aprendizaje, al brindarles realimentación continua sobre su progreso en los proyectos, y al proporcionarles conocimiento aplicado para mejorar en su trabajo, se logró crear un entorno de aprendizaje totalmente transparente y flexible que les generó mayor motivación mediante un aprendizaje más rápido y eficiente.

De manera similar, Carnicero et al., (2021) argumentan que el uso de herramientas de digitalización como las TI o el internet de las cosas, las cuales convergen en la evolución de los distintos agentes educativos y de las nuevas formas de interacción síncrona y asíncrona; la incorporación de distintos métodos ágiles como XP o Scrum; el uso de herramientas de simulación como la realidad virtual, la realidad aumentada, los simuladores, la gamificación, entre otras, pueden generar una mayor motivación entre los estudiantes y lograr una mejor asimilación de conceptos teóricos. De esta manera, la formación académica de los futuros ingenieros alcanza un nivel más práctico, personalizado, y realista, contribuyendo a mejorar el desarrollo de sus capacidades individuales y de trabajo en equipo dentro de un entorno lúdico.

Por lo tanto, se puede considerar que los métodos ágiles han demostrado ser de gran utilidad para la gestión de equipos de trabajo y proyectos de desarrollo de software en el contexto de la actual Industria 4.0, por lo que se han estado adaptando también al contexto educativo no solo en el área de la Ingeniería de Software, sino también en otras áreas mucho más formales como las matemáticas (Duvall et al., 2017). En este contexto, se ha buscado lograr una mayor motivación en los estudiantes universitarios empleando técnicas de aprendizaje que se basen en experiencias, tal es el caso de los juegos. Gracias a este novedoso método de formación denominado “juegos serios” (Deterding et al., 2011), se obtiene un mejor aprendizaje que cuando se estudia de manera tradicional, puesto que genera que los estudiantes tengan una mejora en la productividad y en determinadas capacidades para desarrollar software conforme a la demanda industrial, además de que se logra un reforzamiento en la conducta del estudiante al enfrentarlo a un problema real pero de forma diferente a la tradicional (Rodríguez et al., 2019).

En este sentido, Muñoz et al., (2018) señalaron que existía un desafío importante para el área de la Ingeniería de Software que debía abordarse desde el contexto educativo: que las universidades tuvieran la capacidad de satisfacer las demandas de la industria en cuanto a la formación de personal de alta calidad para el desarrollo de software, mediante la capacitación de los estudiantes en estándares del proceso de software. De manera similar, García et al. (2020) argumentaron que, además de las habilidades de programación; trabajo en equipo; y liderazgo, la mala preparación académica se debe a que los estudiantes universitarios no se familiarizan durante sus estudios con los modelos de procesos de software ni con los estándares del proceso de software, lo que limita con frecuencia sus oportunidades de participar en actividades estratégicas en la industria y, por consiguiente, de conseguir mejores salarios.

En base a lo anterior, se puede observar que en la literatura actual existe evidencia sólida de diferentes propuestas enfocadas en juegos serios para facilitar el aprendizaje de un estándar del proceso de software. Por ejemplo, Calderón et al. (2018a) propusieron mejorar la educación de los

estudiantes universitarios y profesionales del desarrollo de software en tópicos de la gestión de los proyectos de software mediante el desarrollo de un juego serio llamado “ProDec”, el cual se basa en la simulación para facilitar la comprensión y enseñanza del estándar ISO/IEC 21500. Los resultados obtenidos evidenciaron que dicho juego fue capaz de cubrir 7 de los 10 grupos temáticos de dicho estándar.

De manera similar, García et al. (2020) propusieron un entorno de aprendizaje interactivo que facilita la introducción al estándar ISO/IEC/IEEE 29148:2018 con un juego serio de nombre “Requengin”, el cual se basa en la simulación y tiene como objetivo principal reforzar la comprensión y la aplicación de los principales procesos definidos por dicho estándar. Los resultados obtenidos fueron favorables, puesto que indicaron que dicho juego serio contribuyó en mejorar la adquisición de conocimientos del estándar ISO/IEC/IEEE 29148:2018 y mejorar la motivación por parte de los estudiantes.

Por lo tanto, es evidente la creación de propuestas para apoyar el aprendizaje de los estándares del proceso de software a nivel universitario. No obstante, el interés de esta tesis es un estándar en específico, el estándar ISO/IEC/IEEE 26515:2018 *Systems and software engineering — Developing information for users in an agile environment*, como se explica a continuación.

### 1.3. Necesidad de la solución

La interfaz de un software a menudo debe ser intuitiva para la mayoría de los usuarios o, al menos, debe apegarse a convenciones comunes que limiten la necesidad de información exhaustiva o sumamente detallada. No obstante, los usuarios deberían recibir información precisa que les ayude en dado caso de que tuvieran la necesidad de saber cómo usar el software. Dicha información debería explicar las principales características implementadas para satisfacer las necesidades de los usuarios ya que la información bien diseñada no solamente ayuda a los usuarios, sino que también reduce costos de capacitación y soporte, y mejora la calificación del producto, de su productor y de sus proveedores. Aunque los métodos de desarrollo ágiles a menudo abogan por la generación de menos documentación durante el ciclo de vida, los usuarios del software aún esperan y requieren que se les proporcione información de calidad sobre el producto a pesar de que, aunque los resultados finales de crear información útil para los usuarios son los mismos, los métodos pueden ser muy diferentes en un entorno de desarrollo ágil. En este sentido, el estándar ISO/IEC/IEEE 26515:2018 *Systems and software engineering — Developing information for users in an agile environment* (2018) proporciona puntualmente requisitos para que los roles correspondientes desarrollen información de calidad en proyectos que utilizan métodos ágiles de desarrollo. De hecho, la documentación del estándar ISO/IEC/IEEE 26515:2018 considera a tres procesos principales:

- Desarrollo de la información, en este proceso se siguen utilizando las fases básicas (e.g., análisis, diseño, programación, revisión) que se describen en el estándar ISO/IEC 26514:2008 *Systems and software engineering — Requirements for designers and developers of user documentation* (2010). Sin embargo, debido a que un requisito para adoptar el estándar ISO/IEC/IEEE 26515:2018 es el uso de las prácticas ágiles, se elimina la clara separación de fases secuenciales del proceso en una o más iteraciones y se reemplaza por un desarrollo con iteraciones cortas.
- Gestión del desarrollo de la información, en este proceso se describen los tipos de documentación que deben desarrollarse en un entorno de desarrollo ágil. Los documentos típicos pueden incluir historias de usuarios, casos de uso, descripciones de personas, listas de tareas, informes Scrum y más. De hecho, el estándar ISO/IEC/IEEE 26515:2018 describe las

actividades que ocurren durante el proceso ágil para garantizar que se desarrolle información de alta calidad. Por lo tanto, el estándar no descuida la obtención de comentarios de los usuarios, algo que con frecuencia se ignora en muchos proyectos ágiles.

- Preparación de la información para los usuarios, en este proceso se establece que los requisitos de los usuarios son la base de toda la información producida en el proyecto ágil de desarrollo. Estos requisitos deberán registrarse para que los planes del proyecto y los planes de desarrollo de la información los tomen como referencia, permitiendo que las modificaciones o las características nuevas puedan rastrearse hasta los usuarios.

De acuerdo con Torres (2020), la variedad y el dinamismo en la industria actual de software han introducido una tendencia por desarrollar software en menos tiempo, mejorando al mismo tiempo la calidad de los productos, y reduciendo los costos para llegar a más mercados. Sin embargo, uno de los principales inconvenientes de las metodologías tradicionales para desarrollar software es que la información que se crea para el usuario no siempre es utilizada y genera un gasto importante debido al nivel de detalle y la cantidad excesiva de documentación que se elabora. Este problema se solventa con los métodos ágiles, las cuales dan mayor valor al software en funcionamiento que a la documentación excesiva (Gamboa, 2018). No obstante, los usuarios finales aún esperan recibir información de calidad con cada producto de software que adquieren. Debido a esto, el estándar ISO/IEC/IEEE 26515:2018 establece los requisitos y brinda la orientación necesaria para crear información de calidad para los usuarios de software que es desarrollado con un enfoque ágil. En este sentido, durante los últimos años se ha observado la continua adopción del estándar ISO/IEC/IEEE 26515:2018 dentro de la industria moderna de software. Kamthan y Shahmir (2016), por ejemplo, propusieron la conjunción de modelos conceptuales para obtener historias de usuario mediante un proceso centrado en el cliente. Dichas historias se basaron en la experiencia del usuario, y el proceso buscaba brindar realimentación constante al cliente con el objetivo de que éste tuviera una experiencia positiva durante todo el proceso de desarrollo. Las historias de usuario que se generaban con dicho proceso eran compatibles con una amplia variedad de métodos ágiles (e.g., Scrum, XP) y con la norma ISO/IEC/IEEE 26515:2018.

De manera similar, Avdeenko y Murtazina (2018) generaron una propuesta de soporte al proceso de gestión de requisitos en un entorno ágil que consideraba la correcta documentación de las historias de usuario para obtener requisitos funcionales. La documentación de tales historias también se basó en el estándar ISO/IEC/IEEE 26515:2018, lo que permitió que los integrantes de un equipo ágil de desarrollo tuvieran una mejor comprensión de la actividad de documentación, identificaran los objetivos comerciales del producto, establecieran correctamente las pruebas de aceptación requeridas para el producto, y desarrollaran documentación de calidad para los usuarios.

En este mismo contexto, Mathrani et al., (2021) realizaron un estudio sobre las prácticas utilizadas para crear la documentación en Scrum con el objetivo de lograr el cumplimiento de la norma ISO 9001. Dicha investigación evidenció el descontento del personal de una empresa de desarrollo de software puesto que dicha norma requería la generación de documentación exhaustiva que a menudo estaba incompleta o era incorrecta y, por consiguiente, se tenía la impresión de que se estaban perdiendo los beneficios del desarrollo ágil. Para abordar estos problemas se sugirió hacer uso de cláusulas específicas del estándar ISO/IEC/IEEE 26515:2018 con el objetivo de contrarrestar dichos problemas y lograr una coexistencia entre el estándar ISO 9001 y los enfoques ágiles utilizados para el desarrollo de software.

Por otro lado, en la actualidad existen diversas propuestas para apoyar la enseñanza de los enfoques ágiles a nivel universitario. De hecho, Rodríguez et al., (2019) y Salza et al., (2019) han argumentado que las entidades educativas debían integrar la enseñanza de los métodos ágiles en sus planes de estudio con el objetivo de brindarle conocimiento especializado a sus estudiantes. Aunado a esto, se planteó la necesidad de combinar técnicas de aprendizaje con herramientas digitales actuales basadas en la experiencia con el objetivo de mejorar la motivación y la productividad de los estudiantes, además de reforzar sus conocimientos y habilidades sociales para resolver problemas en entornos reales de desarrollo ágil. Rodríguez et al. (2020), por ejemplo, presentaron un estudio sistemático sobre diez juegos serios creados para mejorar la enseñanza de distintos métodos ágiles, los cuales reflejaron resultados positivos en cuanto a la efectividad del aprendizaje y al aumento del compromiso y la motivación de los estudiantes. En este mismo sentido, Carnicero et al., (2021) afirman que el uso de estos métodos, como apoyo en la educación de los estudiantes, brinda ventajas significativas en su formación académica y ayuda a mejorar la comprensión de los conceptos y las distintas prácticas ágiles.

Sin embargo, a pesar de que existen muchas propuestas creadas para mejorar la enseñanza de un proceso de desarrollo ágil de software, hasta la presentación de esta tesis no se encontraron propuestas enfocadas al estándar ISO/IEC/IEEE 26515:2018, situación que deja en claro que es importante generar una propuesta que complete el aprendizaje de los métodos ágiles con el objetivo de generar documentación de calidad a lo largo de un ciclo de desarrollo.

Por lo tanto, considerando todo lo anteriormente expuesto, esta tesis pretendió desarrollar un juego serio para apoyar la educación, mejorar la comprensión, fortalecer el aprendizaje, y fomentar una enseñanza más activa del estándar ISO/IEC/IEEE 26515:2018 *Systems and software engineering —Developing information for users in an agile environment*, con el objetivo de contribuir a mejorar la preparación de los estudiantes universitarios familiarizándolos con uno de los estándares más utilizados actualmente en la industria.

## 1.4. Delimitaciones de la tesis

Esta tesis estuvo sujeta a las siguientes delimitaciones y limitaciones:

- El juego serio cubrió únicamente los procesos y actividades descritos en el estándar ISO/IEC/IEEE 26515:2018 *Systems and software engineering —Developing information for users in an agile environment*.
- El juego serio se creó únicamente para el sistema operativo Windows de Microsoft®.
- Se decidió que la experimentación solamente incluyera a estudiantes universitarios de carreras afines a la Ingeniería de la Computación que incluyeran la asignatura de “Ingeniería de Software” en sus planes de estudio. Se trabajó preferentemente con estudiantes de noveno semestre de la Universidad Tecnológica de la Mixteca (UTM).
- Se planteó al inicio que los resultados de esta tesis podrían verse afectados por el brote de una nueva pandemia que no permitiera llevar a cabo un caso de estudio de manera presencial. Afortunadamente esto no ocurrió.
- Los resultados de la experimentación pudieron verse afectados negativamente por la baja motivación o disponibilidad de los profesores y estudiantes al aplicar un paradigma alternativo de aprendizaje.

## 1.5. Hipótesis del trabajo

La hipótesis de esta tesis se planteó como una relación causal y se enunció de la siguiente manera:

“El uso de un juego serio en el aprendizaje del estándar ISO/IEC/IEEE 26515:2018 *Systems and software engineering —Developing information for users in an agile environment*, mejorará la motivación, satisfacción, y el desempeño de estudiantes de nivel licenciatura”.

## 1.6. Objetivos del trabajo

Los objetivos de la presente tesis se describen a continuación:

### 1.6.1. Objetivo general

***“Desarrollar un juego serio que fortalezca la enseñanza del estándar ISO/IEC/IEEE 26515:2018 Systems and software engineering — Developing information for users in an agile environment entre estudiantes de nivel licenciatura”.***

### 1.6.2. Objetivos específicos

- Realizar un análisis integral de propuestas similares que hayan sido desarrolladas con el objetivo de facilitar la enseñanza del mencionado estándar.
- A partir de los estudios encontrados, identificar aspectos importantes que pudieran ser considerados en la elaboración del juego serio propuesto en esta tesis.
- Desarrollar un método de aprendizaje dinámico para la enseñanza de los componentes más importantes de dicho estándar. Este método se compondrá de dinámicas y mecánicas de juego que facilitarán el aprendizaje, sin que los estudiantes necesariamente sepan que están aprendiendo conceptos de un estándar de la Ingeniería de Software.
- Diseñar y conducir un estudio cualitativo/cuantitativo para recoger la opinión de los estudiantes y profesores sobre el juego serio propuesto como herramienta de soporte a la enseñanza de dicho estándar. Los resultados recogidos se utilizarán para validar la hipótesis establecida anteriormente.

## 1.7. Metodología

De acuerdo con Cico et al. (2021), la mayoría de los estudiantes universitarios carecen de las habilidades duras y suaves que la industria de software requiere en la actualidad, situación que les complica la obtención de un puesto de trabajo bien remunerado. Por lo tanto, las evaluaciones para determinar la “calidad” en la adquisición de competencias y el aprendizaje especializado de los estudiantes no pueden depender únicamente de exámenes convencionales, ya que es necesario evaluar otras actividades complementarias que refuercen la adquisición de habilidades. En este sentido, para reducir la brecha existente entre conocimiento y habilidades, se han generado múltiples propuestas educativas, como los juegos serios. De acuerdo con Kühn (2019), un juego serio utiliza el entretenimiento para complementar la formación del estudiante empleando los principios de diseño de juegos para fines distintos a los lúdicos, favoreciendo así el desarrollo de las competencias y mejorando de forma eficaz el aprendizaje y el desarrollo de habilidades específicas.

Es verdad que los juegos serios han demostrado resultados positivos al ser utilizados como un complemento en la educación de la Ingeniería de Software. No obstante, existen otras técnicas como la “gamificación” que toma elementos clave de los juegos para fomentar, por ejemplo, el comportamiento positivo entre el personal de organizaciones enfocadas al desarrollo de software, sin la necesidad de desarrollar un juego serio, y también han mostrado resultados favorables en el apoyo al aprendizaje. En este sentido, durante los últimos años se han realizado estudios que respaldan el uso de esta técnica en un contexto industrial. Por ejemplo, Stol et al., (2022) publicaron un estudio empírico de una organización que desarrolló una plataforma de gamificación para estimular el comportamiento y mejorar la participación de los desarrolladores de software. Dicho estudio concluyó que la gamificación puede utilizarse como un instrumento eficiente para involucrar y motivar al personal y obtener, como consecuencia, una mejor participación dentro de la organización. Además, se argumenta que el modelo teórico propuesto para lograr tal fin “postula que el deseo de los desarrolladores de aprender nuevas habilidades y tecnologías se relaciona positivamente con la participación en desafíos de gamificación, pero que esta relación está moderada por la experiencia del desarrollador”. Sin embargo, también es verdad que un empleado de la industria de software no dispone del mismo tiempo que un estudiante para jugar y aprender o fortalecer sus conocimientos, por lo que un juego serio podría dar mejores resultados en un contexto académico. De hecho, considerando lo anterior queda patente la diferencia entre gamificación y juego serio, puesto que la gamificación implica un juego parcial (principalmente dinámicas), mientras que un juego serio implica un juego completo, valga la redundancia, que debe establecer objetivos de aprendizaje para alcanzar un objetivo educativo.

En este sentido, Marín (2022) presentó un estudio que apoya esta afirmación. Dicho estudio presentó la evaluación de cinco juegos serios (RISE, Scrumption, BPMN, Classutopia, y BTP) utilizados para promover el aprendizaje activo de diferentes temas de la Ingeniería de Software. Los autores concluyeron que, si bien la gamificación tiene el potencial necesario para mejorar el aprendizaje entre los estudiantes, los juegos serios pueden lograr una mayor motivación en los estudiantes, lo que se traducirá en una mejor comprensión de los temas que conducirá a la obtención de conocimientos más sólidos.

Considerando lo anterior, esta tesis propuso el desarrollo de un juego serio para apoyar la educación, mejorar la comprensión, fortalecer el aprendizaje, y fomentar una enseñanza más dinámica del estándar ISO/IEC/IEEE 26515:2018 *Systems and software engineering —Developing information for users in an agile environment*, con el objetivo de contribuir a mejorar la preparación de los estudiantes universitarios familiarizándolos con la aplicación correcta de uno de los estándares más utilizados actualmente en la industria para la documentación en entornos de desarrollo ágil. Por lo tanto, el desarrollo de la solución planteada en esta tesis consistió en dos etapas que son descritas a continuación.

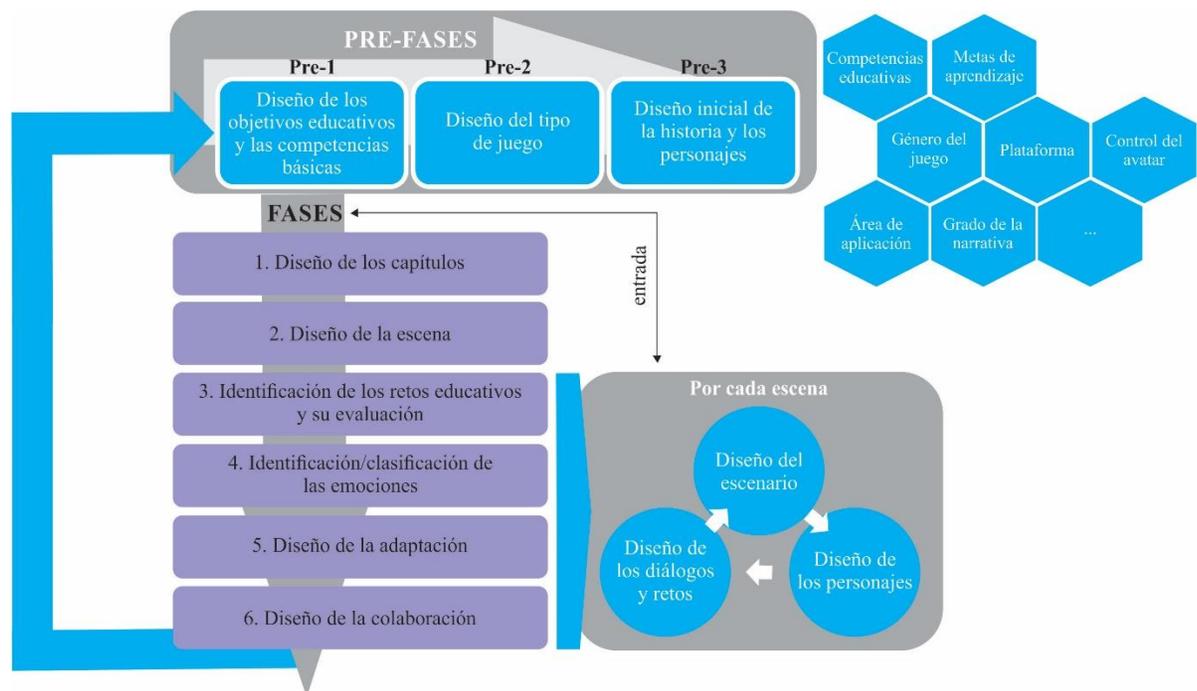
### **1.7.1. Diseño e implementación de un juego serio**

El juego fue implementado considerando la metodología basada en guiones interactivos propuesta por de Lope et al., (2015). La Figura 1.1 muestra que dicha metodología establece una serie ordenada de pasos iterativos que inician con tres fases preliminares.

Las fases preliminares de la metodología son las siguientes:

- *Pre-1: Diseño de los objetivos educativos y las competencias básicas.* Se determinan los objetivos educativos y las competencias que serán abordadas en el juego.

- *Pre-2: Diseño del tipo de juego.* Se determinan las características del juego que pudieran influir en las decisiones futuras de diseño.
- *Pre-3: Diseño inicial de la historia y los personajes.* Se desarrolla la historia a través de iteraciones que faciliten el entendimiento de las mecánicas y dinámicas pensadas para el juego.



**Figura 1.1.** Metodología de desarrollo de juegos basada en guiones interactivos (de Lope et al., (2015), como se citó en (León, 2016))

Una vez que las fases preliminares fueron concluidas, se deben desarrollar las siguientes fases:

- **Fase 1: Diseño de los capítulos.** Un juego serio debe contener al menos un capítulo que se defina por un identificador, un nombre abreviado, y la trama general de la aventura que se desarrollará.
- **Fase 2: Diseño de la escena.** Cada capítulo debe dividirse en escenas que involucren el diseño de los escenarios, personajes, y diálogos y desafíos del juego.
- **Fase 3: Identificación de los retos educativos y su evaluación.** Los objetivos educativos a menudo se relacionan con los retos lúdicos diseñados para el juego. En este sentido, será necesario crear mecanismos y reglas de evaluación para las tareas educativas diseñadas.
- **Fase 4: Identificación/clasificación de las emociones.** Las emociones consideradas deben relacionarse con las acciones que los jugadores realizarán en el juego con el fin de abordar un objetivo educativo.
- **Fase 5: Diseño de la adaptación.** Durante esta fase es obligatorio determinar si el juego tiene la capacidad de adaptarse a las capacidades y características del jugador, del dispositivo de juego, o del entorno de juego.

- **Fase 6: Diseño de la colaboración.** En la última fase es obligatorio indicar qué acciones o serie de pasos, si las hubiera, deberán realizarse en grupo.

Considerando dicha metodología, se pretendió crear un juego serio para la enseñanza de los procesos y conceptos básicos del estándar ISO/IEC/IEEE 26515:2018, enfocándose en estudiantes universitarios de carreras afines a la Ingeniería de Software. En este sentido, es importante mencionar que este trabajo no pretendió sustituir los métodos de enseñanza tradicionales, sino crear una herramienta de apoyo educativo que los complementa.

### 1.7.2. Realización de una evaluación empírica

La segunda etapa del desarrollo de la tesis se centró en la realización de un caso de estudio que permitiera la recolección de información para validar la hipótesis planteada en esta tesis. Con este objetivo en mente se propuso el diseño de un estudio cualitativo/cuantitativo que incluyera lo siguiente:

#### 1.7.2.1. Participantes

Dicha evaluación empírica incluyó a estudiantes de nivel universitario de noveno semestre de la carrera de Ingeniería en Computación de la UTM, de ambos géneros, y edades entre 21 y 24 años. Los estudiantes fueron divididos en dos grupos de igual cantidad de integrantes, el primer grupo fue denominado “grupo de control” y el segundo fue denominado “grupo experimental”. El caso de estudio no distinguió o evaluó variaciones entre el desempeño de mujeres y/o hombres. De igual manera, se contó con la participación de profesores quienes supervisaron a los grupos de estudiantes que se formaron para participar en el estudio, de forma que siempre hubo un responsable por cada grupo.

#### 1.7.2.2. Instrumentos

La evaluación requirió de los siguientes instrumentos:

- Se utilizó el Modelo para la Evaluación de Juegos Educativos enfocados en la Enseñanza de la Computación (MEEGA+, por sus siglas en inglés) (Petri et al., 2019) con el objetivo de determinar la motivación, experiencia de uso, y aprendizaje de los estudiantes. Dicho modelo define un cuestionario de 27 preguntas que abordan las dimensiones del aprendizaje a corto y largo plazo propuestas por Moody y Sindre (2003) y la escala de motivación de materiales instruccionales (Keller, 2009).
- Por otra parte, también se siguieron las recomendaciones de Bourgonjon et al., (2013) y Bodnar et al., (2015) para diseñar un cuestionario que evaluara la percepción de los profesores sobre el juego serio propuesto. Dicho cuestionario está conformado por tres partes: la información demográfica como la edad y género de los profesores; las variables relacionadas con el profesorado como sus años de experiencia, asignaturas que imparte, el grado escolar; y la postura de los profesores referente al uso del juego serio como recurso educativo. La escala usada en estos cuestionarios es tipo Likert de cinco puntos que va desde -2 (“Totalmente en desacuerdo”) hasta 2 (“Totalmente de acuerdo”).
- Aunado a lo anterior, se diseñó un mecanismo de diagnóstico que evaluó los conocimientos teóricos/prácticos sobre el estándar ISO/IEC/IEEE 26515:2018, el cual fue aplicado antes (pre-evaluación) y después (post-evaluación) de la sesión de juego con el fin de obtener la

información necesaria para realizar una comparación cuantitativa sobre el aprendizaje de los estudiantes participantes del estudio.

- Finalmente, se realizó un análisis estadístico de toda la información recogida en el estudio con el objetivo de explicar las diferencias encontradas entre ambos grupos de estudiantes.

### 1.7.2.3. Proceso de evaluación

El proceso experimental siguió la siguiente lógica:

- Se inició con la impartición tradicional de clases teóricas sobre el estándar ISO/IEC/IEEE 26515:2018, por lo que el(la) profesor(a) utilizó cualquier material didáctico que fuera de utilidad en el formato tradicional de enseñanza. Aunado a lo anterior, parte de esta teoría se enfocó a enseñar las principales herramientas, métodos, y/o técnicas útiles para el desarrollo de la información de un proyecto que es elaborado en un entorno ágil.
- Una vez que los estudiantes terminaron esta formación teórica, fueron separados en un grupo de control y otro experimental con igual número de participantes. Los estudiantes del grupo de control realizaron prácticas de la manera tradicional sobre los principales conceptos del estándar ISO/IEC/IEEE 26515:2018, mientras que los estudiantes del grupo experimental utilizaron el juego serio creado para tal fin. Hasta este punto se concluyó con la participación de los estudiantes del grupo de control, quienes resolvieron un examen de conocimientos sobre el estándar.
- Por lo tanto, se asignaron sesiones de juego para que los estudiantes del grupo experimental tuvieran la oportunidad de aplicar el conocimiento adquirido durante las clases teóricas y, al mismo tiempo, lograran familiarizarse, de una forma más dinámica, con uno de los estándares del proceso de software más utilizados en la industria. Para lograr dicho objetivo, los estudiantes de este grupo atravesaron las diferentes escenas y escenarios, cumplieron los retos y misiones, y respondieron preguntas que se les presentaron en relación con el manejo y adecuación de la información de usuario en entornos ágiles de desarrollo.
- Con el apoyo de esta herramienta el(la) profesor(a) determinó si los estudiantes comprendieron correctamente los conceptos básicos del estándar ISO/IEC/IEEE 26515:2018 y, al mismo tiempo, analizó de manera directa sus diferentes reacciones y determinó los niveles de motivación y participación. Por otro lado, el(la) profesor(a) pudo monitorear qué procesos o conceptos no fueron correctamente asimilados, y dieron la oportunidad a los estudiantes de reforzarlos mediante la realización de competencias con estudiantes más adelantados.
- Se realizaron dos evaluaciones a los estudiantes del grupo experimental. En primera instancia se aplicó el mismo examen de conocimientos que fue aplicados a los estudiantes del grupo de control para medir su nivel de aprovechamiento sobre los conceptos teóricos aprendidos. Posteriormente, se utilizó el MEEGA+ (Petri et al., 2019) con el objetivo de medir la motivación y satisfacción de estos participantes.
- Finalmente, se utilizó el cuestionario diseñado siguiendo las recomendaciones de Bourgonjon et al., (2013) y Bodnar et al., (2015) para evaluar la percepción de los(as) profesores(as) sobre el juego serio propuesto.
- Para concluir la experimentación, todos los datos recogidos durante la evaluación fueron analizados estadísticamente para identificar diferencias notables entre ambos grupos de estudiantes.

## 1.8. Estructura de la tesis

En el Capítulo 2 se presenta el estado del arte de la investigación actual relacionada con el tópico abordado en esta tesis. En este sentido, se comienza abordando los antecedentes de la investigación, posteriormente se presentan los fundamentos del estándar ISO/IEC/IEEE 26515:2018 *Systems and software engineering —Developing information for users in an agile environment*, para finalmente definir la importancia de su enseñanza en el contexto de la industria actual de software. Por último, se analizan trabajos similares al propuesto en esta tesis con el objetivo de establecer una base comparativa.

En el Capítulo 3, la parte más extensa de la tesis, se describe paso a paso la metodología seguida para diseñar y construir el juego serio propuesto como solución al problema identificado.

El Capítulo 4 presenta el diseño y realización de un caso de estudio para evaluar empíricamente el juego desarrollado. Por lo tanto, se describen los resultados y las lecciones aprendidas con el uso de dicho juego. Finalmente, se resumen las amenazas a la validez de la evaluación realizada.

En el Capítulo 5 se presentan las principales conclusiones derivadas de la investigación, algunas conclusiones personales, y una propuesta de trabajo futuro.

Finalmente, la sección de Bibliografía presenta las referencias bibliográficas base utilizadas para el desarrollo de la tesis.

En la sección de Anexos se presenta cualquier información adicional requerida para realizar este trabajo de tesis.

## **2. Marco teórico**

En el segundo capítulo de esta tesis se exponen todos los conceptos, argumentos, y teorías en las cuales se sustenta la presente investigación. Además, se pretende brindar un análisis del estándar ISO/IEC/IEEE 26515:2018 *Systems and software engineering — Developing information for users in an agile environment*, con el objetivo de sintetizar las partes que lo componen e identificar cuáles de ellas resultan cruciales para ser abordadas mediante una propuesta educativa. Por último, se describen propuestas similares de juegos serios que fueron creados con el mismo objetivo de contribuir a mejorar la enseñanza/aprendizaje de los estándares del proceso de software.

### **2.1. Uso de estándares en la industria de software**

En la actualidad, el software está presente en casi todas las actividades que se realizan en la vida cotidiana yendo desde las aplicaciones que se utilizan en los teléfonos celulares y las computadoras, hasta los sistemas que influyen directamente en el bienestar de la sociedad. En este sentido, la exposición del software cambió drásticamente, puesto que ya no es de uso exclusivo de los desarrolladores de software o científicos, sino que ahora usuarios de todas las edades pueden utilizarlo para resolver problemas triviales y/o complejos. Sin embargo, de acuerdo con Stephens (2015), esta situación también generó un cambio importante en la forma en la que el software era desarrollado puesto que se generaron nuevas necesidades en el mercado tecnológico que demandaban mejores productos con mayor calidad, pero de menor costo. Pero la necesidad de cubrir estas demandas de la industria emergente ocasionó que el software se volviera cada vez más complejo de desarrollar y que su costo se incrementará considerablemente. Por lo tanto, la Ingeniería de Software juega un papel relevante en este escenario. La Ingeniería de Software es la disciplina que define a todas las fases requeridas en la producción de software, desde la etapa preliminar de recopilación de requisitos hasta el mantenimiento y soporte del sistema de software. En comparación con otras disciplinas tradicionales de la ingeniería, se puede resaltar la rápida evolución de la Ingeniería de Software ocasionada por la necesidad de crear sistemas de gran tamaño y complejidad mediante procesos eficientes de desarrollo. Es decir, existen ramas de la ingeniería como la Mecánica, Química, o Física que se basan totalmente en las leyes de la naturaleza; sin embargo, la Ingeniería de Software se basa únicamente en las leyes de la lógica y las matemáticas. Desafortunadamente, de acuerdo con Dauda et al. (2021), el desarrollo de software a menudo se ve afectado por varios desafíos, como pueden ser los contextos situacionales variables, el cambio constante de los requisitos durante todo el proceso de desarrollo, la optimización y presión del cronograma, la complejidad e invisibilidad del software, entre muchos más. Por lo tanto, diversos organismos han creado guías con el objetivo de definir procesos eficientes para desarrollar software, a las cuales se les denominó estándares del proceso de software.

El diccionario Merriam-Webster argumenta que el término “estándar” hace referencia a *“lineamientos establecidos por una autoridad, costumbres, o el consentimiento general como modelo o ejemplo o (...) como regla para la medida de una cantidad, peso, extensión, valor, o calidad”*. De acuerdo con Laporte y April (2018), la creación y aceptación de un estándar de este tipo requiere de un proceso riguroso que refleja el consenso de una comunidad de especialistas. Los estándares son utilizados por las organizaciones gubernamentales, por ejemplo, como documentos cuasi legales debido a que a menudo son requeridos para probar o refutar elementos en un tribunal de justicia sirviendo como un requisito legal. Cuando esto ocurre, el contenido del estándar es de suma importancia ya que las organizaciones que lo utilizan para desarrollar productos y/o servicios pudieran generar impactos en la vida humana, medio ambiente, o negocios. En el contexto de la Ingeniería de Software, la adopción de un estándar se considera una forma de transferencia de tecnología, por lo que es necesario hacer una correcta selección del mismo y realizar una aplicación adecuada al contexto en el que se pretende utilizar con el fin de aumentar la productividad de una organización (Laporte et al., 2018). Dauda et al., (2021) argumentan que los estándares privilegian la comunicación y el entendimiento compartido sobre todas las cosas, puesto que reconocen las diferencias de los contextos operativos y situacionales de las organizaciones al definir un ciclo de vida para el desarrollo de software, produciendo múltiples beneficios al usarlos. Aun así, uno de los retos importantes para la Ingeniería de Software es que los estándares que contienen conocimientos especializados y prácticas documentadas de éxito lleguen a entenderse realmente dentro de la organización con el objetivo de obtener tales beneficios.

En este contexto, se ha desarrollado una cantidad importante de estándares del proceso de software para tratar de ayudar a estas organizaciones con la difícil tarea de crear productos de software que cubran la mayoría de las necesidades de los clientes potenciales. Sin embargo, dada la complejidad del proceso de desarrollo de software, se han desarrollado estándares para prácticamente cada una de las fases de su desarrollo por lo que es prácticamente imposible encontrar un solo estándar que cubra todos los requerimientos de un mismo producto, proceso, e incluso tipo de organización. Por ejemplo, Ramírez et al. (2020) realizaron un estudio para comparar distintos estándares que soportan el desarrollo de software seguro, llegando a la conclusión de que no es posible cubrir todos los requisitos de seguridad en el desarrollo de software basándose en un solo estándar, ya que es necesaria la implementación de más de uno para cumplir con la regulación del mercado. De igual modo, Masuda et al. (2020) presentaron un método de análisis de pruebas para el software complejo haciendo uso de definiciones de arquitectura de software presentes en diversos estándares de programación, debido a que los estándares para las pruebas de software son difíciles de usar.

Así pues, el Comité de Estándares de Ingeniería de Sistemas y Software del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por sus siglas en inglés) es el organismo que se ha establecido durante cuatro décadas como la máxima autoridad en el desarrollo de estándares para apoyar a los especialistas, administrando la dirección y el alcance de la ingeniería y estándares de software y sistemas con el objetivo de mejorar la eficacia y eficiencia en los procesos de la Ingeniería de Software, mejorar las comunicaciones entre proveedores y adquirientes, y mejorar la calidad de los procesos y productos de software (Rosdahl, 2018). Este mismo comité ha reconocido que existe un debate importante sobre la aplicabilidad y uso de estos estándares dentro de la industria de software puesto que las organizaciones que la conforman tienen características diferentes. Por ejemplo, el estudio de O’Connor (2019) señaló que al hacer uso de estándares, particularmente en las pequeñas y muy pequeñas organizaciones de software donde la comunicación y las personas tienen mayor importancia, se observa una mejoría en la calidad del producto de software, mayor competitividad, mayor confianza y satisfacción del cliente, mayor patrocinio para la mejora de procesos y un menor riesgo de fracaso durante el desarrollo, además de que facilitan la comercialización potenciando así la

exportación del producto de software. En este mismo sentido, Franch et al. (2021) también argumentaron que el cumplimiento de estándares en la Ingeniería de Requisitos, por ejemplo, tiene múltiples ventajas, entre las cuales destaca que apearse a ellos ayuda a evitar errores y mejora la calidad de los productos de software que se producen, además, ocasiona que dichos productos sean más fáciles de usar y comprender, especialmente para las personas que no estuvieron involucradas en su desarrollo. Esto genera una menor dependencia de las capacidades individuales y los estilos de trabajo de las personas, permitiendo compartir el conocimiento y la experiencia de otros, especialmente cuando muchos fabricantes, productores y proveedores colaboran e interactúan para desarrollar productos y brindar servicios para diversos dominios. Asimismo, Laporte y April (2018) argumentaron que algunos de los estándares creados para la Ingeniería de Software se ampliaron en cuanto a su alcance llegando a otras áreas como, por ejemplo, a la Ingeniería de Sistemas. Es decir, los estándares de validación y verificación, gestión de riesgos, y gestión de la configuración se han extendido para cubrir el campo de la Ingeniería de Sistemas que se encarga de desarrollar productos que a menudo incluyen hardware, facilitando la comunicación entre contextos distintivos.

A menudo se argumenta que los beneficios que se obtienen al utilizar los estándares del proceso de software son aplicables a todas las organizaciones desarrolladoras de software. Puede que esto sea verdad, pero en México, donde más del 98% de las organizaciones dedicadas al desarrollo de software son pequeñas o muy pequeñas, a menudo se presentan muchas dificultades para relacionar tales estándares con sus necesidades comerciales y justificar su aplicación, por consiguiente no se logran visualizar los beneficios netos de aplicarlos ya que estas organizaciones carecen de experiencia o no tienen los recursos monetarios, de personal o de tiempo, generando así que exista una percepción errónea de que los estándares pueden utilizarse en todo tipo de organizaciones (Laporte y April, 2018). Aunado a lo anterior, en la actualidad existe evidencia de que la mayoría de las organizaciones desarrolladoras de software que son pequeñas y muy pequeñas no adoptan los estándares existentes debido a que perciben de forma errónea que éstos son desarrollados y están orientados exclusivamente para las organizaciones grandes que cuentan con una cantidad importante de empleados especializados (Coleman y O'Connor, 2008a). Por lo tanto, el rechazo generalizado de utilizar los estándares se debe en gran medida a las opiniones negativas sobre el costo, la documentación, y la burocracia de su implementación. Aunado a esto, a las organizaciones pequeñas y muy pequeñas les es difícil destinar los recursos necesarios para establecer los procesos de software apeándose a los estándares actuales (Coleman y O'Connor, 2008b). Si bien es cierto que el hacer uso correcto de los estándares acarrea muchos beneficios, también es verdad que puede perjudicar a la organización si no se toman las medidas necesarias para seleccionar los estándares y procedimientos apropiados para las diversas actividades que se llevarán a cabo durante el desarrollo de software. Además de estos desafíos, la complejidad y la invisibilidad del software son aspectos esenciales de cualquier proceso de desarrollo de software que se pretenda mejorar con el uso de estándares, puesto que el software se va haciendo cada vez más complejo cuando se busca escribir un código eficiente (Dauda et al., 2021).

Considerando lo anterior, desde hace 20 años el desarrollo, la mejora, y la implementación generalizada de los estándares profesionales se ha desarrollado en diversas áreas estableciendo los requisitos de conocimiento, habilidades, competencias, experiencias, valores, y personalidad necesarios para desempeñar una función profesional en el ámbito de la Ingeniería de Software. En este sentido, de acuerdo con Gorbunova et al. (2018), el desarrollo de las Tecnologías de la Información y la Comunicación en el contexto de la Industria 4.0 introdujo nuevas exigencias educativas que demandan graduados mejor calificados. Sin embargo, el estudio realizado por Muñoz et al. (2019) evidenció que los programas académicos actuales no siempre cumplen con los requisitos mínimos para satisfacer las necesidades de la industria actual. Si bien es cierto que, como se mencionó anteriormente, el uso correcto de los estándares puede incrementar la probabilidad de éxito de una

organización al desarrollar software, también es verdad que esto dependerá en gran medida del conocimiento y entrenamiento que reciban los futuros profesionistas (Calderón et al., 2018b). Así, desde hace algunos años se ha atestiguado un mayor interés por desarrollar métodos y técnicas alternativas de aprendizaje en la Ingeniería de Software a nivel universitario, como lo es el uso de los juegos serios (Kosa et al., 2016). De esta manera, se ha promovido el aprendizaje activo e interactivo mediante el aumento de la motivación y el compromiso de los estudiantes universitarios, alcanzando así una formación más integral y dinámica sobre tópicos que requieren de un entrenamiento más práctico (Calderón y Ruiz, 2016). Por lo tanto, las instituciones educativas tienen ahora en los juegos serios una herramienta de formación que permite una interacción continua de los estudiantes con la práctica.

## 2.2. Enseñanza de los estándares en un contexto universitario

De acuerdo con Winters (2022), durante los últimos años se ha observado que la inmensa mayoría de los graduados de carreras profesionales como la Ingeniería en Computación y Licenciatura en Informática frecuentemente deben afrontar varias dificultades al inicio de su vida laboral debido a la falta de coherencia que existe entre las habilidades adquiridas en las aulas universitarias y las habilidades que la industria de software requiere en ese momento. Garousi et al., (2019) argumentaron que esta brecha se podía deber a que la mayoría de los programas universitarios destinan más tiempo a otros tópicos de las Ciencias de la Computación (e.g., usabilidad, redes, compiladores, sistemas operativo, inteligencia artificial), restándole importancia a temas relacionados con la Ingeniería de Software, lo que ha ocasionado que los estudiantes sean formados como científicos informáticos y no reciban conocimiento más específico para que puedan desenvolverse como ingenieros de software. Además, se señala que los conocimientos de los estudiantes recién egresados son limitados como consecuencia de la calidad de su educación puesto que muchas veces no se incluyen tecnologías, técnicas, y/o metodologías modernas en su formación, por lo que es necesario promover el desarrollo generalizado de alternativas educativas. En este sentido, Sleeter y Carmona (2017) sugirieron que una formación integral debería incluir a los estándares que son sustentados por diversas organizaciones, mientras que Calderón, Trinidad, Ruiz y O'Connor (2018) argumentaron que se requiere de un enfoque de enseñanza/aprendizaje más práctico y realista.

Por otro lado, Assyne et al., (2022) afirman que para aumentar la empleabilidad de los estudiantes recién egresados que pretenden incorporarse a la industria de software es necesario mejorar la educación de los mismos. Es decir, se argumenta que los problemas que se presentan en la industria con egresados de reciente contratación no se deben a la escases de conocimientos, sino más bien a la escases de calidad en su formación puesto que *“hacen falta ingenieros bien entrenados y experimentados con una comprensión formal y profunda de la Ingeniería de Software”*. Para lograr esto, Manan y Suryaman (2021) establecen que el material que se utiliza en las universidades con el fin de entrenar mejor a los futuros desarrolladores de software debe considerar las siguientes tareas:

1. Organizar la información del curso de acuerdo con las necesidades que demandan los estudiantes.
2. Formular los objetivos y las metas que se pretenden alcanzar con el desarrollo del curso basándose en las necesidades de formación predeterminadas. Esto servirá como una herramienta para medir el nivel de éxito de la formación.
3. Materializar los resultados de la evaluación generando el contenido del programa. Este material deberá ser relevante tanto para satisfacer las necesidades de los estudiantes como las de la industria.

#### 4. Aplicar los principios del aprendizaje con el material generado.

Es evidente que el realizar estas tareas podría resultar una tarea difícil en el contexto de nuestro país, puesto que difícilmente se le pregunta a un estudiante qué quiere aprender en un curso de especialización. No obstante, el razonamiento de Manan y Suryaman deja en claro algo fundamental: proporcionarles a los estudiantes el conocimiento que la industria requiere. Esta idea coincide con los argumentos de Anicic y Stapic (2022) quienes afirman que “...es necesario brindar a los estudiantes los conocimientos y habilidades necesarias para integrarse a la industria”. De acuerdo con de León-Sigg et al. (2020), dicho conocimiento debe proveerse en las aulas universitarias para que los estudiantes ya egresados sean capaces de desarrollar adecuadamente sus responsabilidades en la industria. En este contexto, actualmente se puede encontrar una cantidad importante de estudios enfocados en la creación de métodos innovadores de enseñanza destinados al desarrollo de habilidades duras y blandas de la Ingeniería de Software; sin embargo, aún se puede visualizar la brecha existente entre las habilidades suaves y duras aprendidas durante una formación universitaria y las habilidades realmente requeridas por la industria.

Como cualquier otra disciplina, la Ingeniería de Software tiene su propio lenguaje para describir situaciones; sin embargo, este lenguaje es útil en entornos académicos, pero crea una barrera de comunicación con otras disciplinas en la industria. Por lo que, para mejorar su enseñanza, debe incluirse material que establezca una forma común de hacer las cosas en los entornos industriales (Calderón, Trinidad, Ruiz y O’Connor, 2018). Con este objetivo en mente es que se han formulado estándares internacionales de software los cuales se basan en el uso de prácticas bien definidas y que brindan un conjunto de procesos para cubrir todo el ciclo de vida del software, definiendo las actividades necesarias para concebir, desarrollar, implementar y mantener un sistema, producto o servicio de software. La investigación realizada por Laporte y Muñoz (2021) indica que los estudiantes que reciben conocimiento sobre los estándares del proceso de software pueden contribuir a mejorar una organización en cinco aspectos fundamentales:

- Valor de la organización: El capital humano que aporta conocimientos sobre tales estándares contribuye a mejorar el desempeño general de la organización.
- Innovación: Los estándares promueven la difusión de la innovación al destacar las ventajas de un producto de software del resto del mercado.
- Transparencia y ética: Los estándares contribuyen a lograr un mejor cumplimiento de las reglas del mercado frente a la competencia. Es decir, considerando las reglas establecidas en el mercado, los estándares facilitan la eliminación de las organizaciones que las incumplan.
- Ámbito internacional: Al promover el desarrollo de los intercambios internacionales, los estándares brindan a las organizaciones una pasarela para exportar sus productos.
- Calidad de productos y servicios: La estandarización otorga a las organizaciones el grado apropiado de control sobre los problemas relacionados con la seguridad y proporciona una garantía real de calidad.

De hecho, el código de ética y práctica profesional de la Ingeniería de Software creado por el IEEE y la Asociación de Maquinaria Computacional (ACM, por sus siglas en inglés), concebido como una norma para la enseñanza y la práctica de la Ingeniería de Software en 1999 y usados todavía en la actualidad, documenta las obligaciones éticas y profesionales de los ingenieros de software destacando en sus principios 3 y 8 el uso de los estándares como se argumenta a continuación:

*“Principio 3 (Producto): Los ingenieros de software deben asegurarse de que sus productos y las modificaciones relacionadas con éstos cumplan con los estándares profesionales más altos*

*posibles. En particular, los ingenieros de software deberán, según corresponda... (3.06) trabajar para seguir los estándares profesionales, cuando estén disponibles, que sean más apropiados para la tarea en cuestión, apartándose de éstos solo cuando esté ética o técnicamente justificado” (Gotterbarn et al., 1999).*

*“Principio 8 (Uno mismo): Los ingenieros de software participarán en el aprendizaje de por vida con respecto a la práctica de su profesión y promoverán un enfoque ético en la misma. En particular, los ingenieros de software se esforzarán continuamente por... (8.05) mejorar su conocimiento sobre los estándares actuales y la ley que rige el software y los documentos relacionados con los que trabajan” (Gotterbarn et al., 1999).*

Por lo tanto, Anicic y Stapic (2022) también argumentan que, debido a la importancia que tales estándares han tomado en la industria, una cantidad importante de estudios remarcan la relevancia de incorporarlos en la formación de estudiantes universitarios, puesto que un estándar del proceso de software representa una fuente esencial de conocimiento codificado para todos los futuros egresados de carreras relacionadas con las Ciencias de la Computación. Meding y Staron (2020) justifican lo anterior al afirmar que la adopción de los estándares del proceso de software plantea un desafío importante debido a que son difíciles de leer y comprender dado su complicado lenguaje técnico, incluso para las partes interesadas para las que fueron creados. Por lo tanto, se requiere de personal capacitado para lograr su integración o adaptación exitosa a las diferentes estructuras, procesos, y actividades de las organizaciones. En este sentido, Laporte y Muñoz (2021) advierten que es urgente que las instituciones universitarias no solamente incorporen este material para mejorar la formación de los futuros ingenieros de software, sino que también capaciten a sus profesores con el objetivo de que sean capaces de transmitir a los estudiantes el conocimiento codificado de los estándares del proceso de software. De acuerdo con estos investigadores, el hecho de no hacerlo debería ser considerado como una mala práctica educativa.

Calderón y Ruiz (2016), Kosa et al. (2016), y Calderón et al. (2018) coinciden en que esta necesidad de transmitir el conocimiento de los estándares del proceso de software a los estudiantes universitarios ha incrementado el interés de muchos investigadores por desarrollar e incorporar métodos y técnicas alternativas para generar una enseñanza más práctica que promueva el aprendizaje activo e interactivo, al mismo tiempo que aumenta la motivación y el compromiso de los estudiantes. Por lo que, de acuerdo con García et al. (2020) y de León-Sigg et al. (2020), el hecho de incluir a los estándares internacionales en la educación universitaria ha demostrado una utilidad importante en la formación de los estudiantes, puesto que se mejora su comprensión dentro de las aulas al mismo tiempo que se les acercan un poco más a la forma en la que se trabaja en la industria real de software.

### **2.3. El estándar ISO/IEC/IEEE 26515:2018 *Systems and software engineering — Developing information for users in an agile environment***

La interfaz de usuario de un software generalmente debe ser intuitiva para la mayoría de los usuarios o debe al menos seguir convenciones comunes que limiten la necesidad de información exhaustiva o detallada para los usuarios. No obstante, los usuarios deben recibir información precisa sobre cómo usar las funcionalidades del software si surgieran dudas. Esta información debe explicar las principales características o casos de uso creados deliberadamente para todo tipo de usuarios. La información debe ser de fácil acceso y estar claramente escrita para permitirle al usuario un aprendizaje rápido y facilitarle el dominio del software, mientras se reduce la ayuda que brinda el servicio de soporte. En este sentido, de acuerdo con el estándar ISO/IEC 26514:2008 (ISO, 2010, pp. iv) la información bien diseñada no solo ayuda a los usuarios, sino que también reduce el costo de capacitación y soporte.

En este sentido, los proyectos que utilizan métodos ágiles de desarrollo se centran en proporcionar entregas rápidas y frecuentes del software. Estos métodos a menudo implican una planificación detallada a corto plazo y la implementación de procesos en paralelo, en lugar de planificar un proyecto completo en distintas fases. Aunque los métodos ágiles de desarrollo a menudo abogan por una menor documentación del ciclo de vida, los usuarios aún esperan y requieren que se les proporcione información de calidad con un producto de software. Aunque los resultados finales del proceso de desarrollo de información para los usuarios son los mismos, los métodos pueden ser muy diferentes en un entorno ágil.

Los métodos ágiles de desarrollo suelen seguir ciclos de desarrollo iterativos cortos basados en los requisitos y la realimentación del cliente. Para cumplir con los requisitos contractuales y proporcionar información valiosa para los usuarios, los entregables de cada iteración incluyen información relacionada con el conjunto de funcionalidades desarrollado durante esa iteración. La cantidad y la calidad de los entregables de información dependen en parte de la integridad y madurez de las características y el diseño del software después de cada iteración, especificado a través de historias de usuarios, tareas, y personas. Los desarrolladores de información y otro personal involucrado en el desarrollo de información para los usuarios deben comprender los procesos y métodos ágiles de desarrollo utilizados por su organización. Esto les permitirá vincularse sin problemas y proporcionar información relevante y útil para los usuarios. Considerando lo anterior, Mathrani et al., (2021) argumentan que, debido a la naturaleza de los métodos ágiles de desarrollo, los medios tradicionales para desarrollar información para los usuarios finales (tanto impresa como en pantalla) no son del todo aplicables. Considerando este contexto, el estándar ISO/IEC/IEEE 26515 (ISO, 2018) fue desarrollado para proporcionar ayuda a las personas responsables de desarrollar la información de usuario de un software que es creado a través de una metodología ágil de desarrollo.

### 2.3.1. Historia del estándar

La Organización Internacional de Normalización (ISO, por sus siglas en inglés) y la Comisión Electrotécnica Internacional (IEC, por sus siglas en inglés) forman el sistema especializado para la normalización mundial. Los organismos nacionales que son miembros de ISO o IEC participan en el desarrollo de estándares Internacionales a través de comités técnicos establecidos por la organización respectiva para tratar campos particulares de actividades técnicas. Los comités técnicos de ISO e IEC colaboran en campos de interés mutuo. Otras organizaciones internacionales, gubernamentales y no gubernamentales, en coordinación con ISO e IEC, también participan en el trabajo. En el campo de la tecnología de la información, ISO e IEC han establecido un comité técnico conjunto, ISO/IEC JTC 1.

La documentación correspondiente al estándar ISO/IEC/IEEE 26515 fue desarrollada por las sociedades del IEEE y los Comités Coordinadores de la Junta de la Asociación de Estándares del IEEE (IEEE-SA, por sus siglas en inglés). El IEEE desarrolla sus estándares a través de un proceso de desarrollo por consenso, aprobado por el Instituto Nacional Estadounidense de Estándares, que reúne a voluntarios que representan diversos puntos de vista e intereses para lograr el producto final. Los voluntarios no son necesariamente miembros del IEEE y colaboran sin recibir compensación alguna. La revisión del 2018 del estándar ISO/IEC/IEEE 26515 fue realizada por el Comité Técnico Conjunto ISO/IEC JTC 1, *Tecnologías de la Información*, Subcomité SC 7, *Ingeniería de Software y Sistemas*, en colaboración con el Comité de Ingeniería de Sistemas y Software de la sociedad informática del IEEE, bajo el acuerdo de cooperación de la Asociación para el Desarrollo de Estándares entre la ISO y el IEEE.

La primera edición del estándar ISO/IEC/IEEE 26515:2018 fue publicada en 2011 y la segunda en 2018, la cual cancela y reemplaza la primera edición del estándar. Los cambios principales de esta segunda edición con respecto a la edición anterior son los siguientes:

- Se realizó una alineación con el uso generalizado de los métodos ágiles para incluir tanto al desarrollo de sistemas como de software.
- Se sustituyó el término “documentación” basada en papel por el término general “información para los usuarios” donde resultó apropiado.
- Se incluyó al desarrollo de información ágil en múltiples equipos y proyectos, especialmente proyectos en situaciones de mantenimiento continuo como DevOps<sup>2</sup>.
- Se realizaron cambios editoriales.
- Se agregaron nuevas definiciones.

### 2.3.2. Descripción del estándar

El estándar ISO/IEC/IEEE 26515:2018, en lo subsecuente llamado simplemente ISO/IEC/IEEE 26515, fue desarrollado para brindar apoyo a los usuarios de los siguientes estándares:

- ISO/IEC/IEEE 15288, *Systems and software engineering — System life cycle processes*.
- ISO/IEC/IEEE 12207, *Systems and software engineering — Software life cycle processes*.
- ISO/IEC 26514:2008, *Systems and software engineering — Requirements for designers and developers of user documentation* (también disponible como IEEE Std 26514-2010, *IEEE standard for adoption of ISO/ IEC 26514:2008, Systems and software engineering — Requirements for designers and developers of user documentation*).
- Otros documentos de la familia de Estándares Internacionales ISO/IEC/IEEE 265NN.

Como se mencionó anteriormente, dicho estándar está diseñado para ser usado por todas las organizaciones que utilizan el desarrollo ágil o que están considerando la idea de desarrollar proyectos siguiendo los métodos ágiles. El estándar ISO/IEC/IEEE 26515 es independiente del tamaño de la organización y de los métodos y herramientas de desarrollo ágil que éstas utilicen para producir el software. No obstante, se espera que la audiencia del estándar tenga experiencia o conocimiento general sobre la información requerida por el usuario (comúnmente llamada “documentación de usuario”), además de que ésta debe estar familiarizada con los procesos ágiles. En este sentido, el estándar brinda una descripción general de los métodos ágiles, aunque no alienta ni desalienta el uso de ninguna metodología ágil en particular. Por lo tanto, el estándar utiliza terminología ágil genérica tanto como es posible.

La documentación del estándar se orienta a los desarrolladores de información y los roles asociados responsables de producir información para los usuarios del software y sistemas desarrollados dentro de un entorno ágil. Por lo tanto, el documento oficial adopta un enfoque estándar de procesos para especificar la forma en que se puede desarrollar la información para los usuarios en proyectos de desarrollo ágil. En este sentido, en el estándar ISO/IEC/IEEE 26515 se siguen aplicando las fases básicas del proceso de creación de la información descritas en el estándar ISO/IEC 26514, como son el *análisis*, el *diseño*, el *desarrollo*, y la *revisión*. Sin embargo, las prácticas ágiles eliminan la clara separación de fases tanto en una sola iteración como en varias iteraciones. Así pues, el estándar ISO/IEC/IEEE 26515 heredó las características del estándar ISO/IEC 26514 en el sentido que éste define un proceso para crear la documentación del usuario que está enfocado en diseñadores y desarrolladores de la documentación de un software. Es decir, dicho proceso describe cómo establecer

---

<sup>2</sup> De acuerdo con Jabbari et al. (2016), DevOps es una metodología de desarrollo destinada a cerrar la brecha entre el desarrollo y las operaciones, enfatizando la comunicación y la colaboración, la integración continua, la garantía de calidad y la entrega con implementación automatizada utilizando un conjunto de prácticas de desarrollo.

qué información necesitan los usuarios, cómo determinar la forma en que dicha información debe presentarse a los usuarios, y cómo diseñar la información y ponerla a disposición de los usuarios. El estándar no se limita a la fase de diseño y desarrollo del ciclo de vida, sino que incluye actividades a lo largo de los procesos de documentación y gestión de la información.

### 2.3.3. Estructura del estándar

De acuerdo con el estándar ISO/IEC/IEEE 26515, las siguientes prácticas ágiles afectan al proceso de creación de la información:

- a) El desarrollo en iteraciones cortas reemplaza a largas fases secuenciales de un proceso.
- b) Los desarrolladores de la información intentan crear información precisa y completa al final de una iteración. Sin embargo, una revisión puede conducir al desarrollo de información adicional en iteraciones posteriores. Por lo tanto, estas personas reciben capacitación para estimar y completar información para los usuarios a tiempo para realizar la liberación de la iteración. Los equipos ágiles autoorganizados reemplazan los roles y contribuciones formales. Es decir:
  - i. Los desarrolladores de software, los *testers*, y los líderes de equipo pueden contribuir al análisis, el diseño, el desarrollo y la revisión de la información para los usuarios.
  - ii. Los desarrolladores de información pueden contribuir al diseño y las pruebas del software, así como a proyectar artefactos y documentos del ciclo de vida, como historias de usuario, casos de uso, y personas.
- c) En proyectos ágiles, los casos de uso y la comunicación oral se utilizan con mayor frecuencia que las especificaciones formales y los documentos de diseño. Por lo tanto:
  - i. Los casos de uso, las historias de usuario, y las personas reflejan claramente el propósito y el beneficio del usuario de cada característica.
  - ii. Las demostraciones y las revisiones por pares validan cómo el desarrollo ha logrado el propósito y el beneficio previstos.
  - iii. Los desarrolladores de información son miembros integrados del equipo ágil para promover una comunicación eficiente y la calidad de los entregables de información.
- d) Los envíos o versiones de características tempranas y frecuentes reemplazan los hitos de lanzamiento programados. Esto significa que los entregables de información deben ser parte de los criterios de aceptación de una liberación funcional completa o un incremento terminado.

El desarrollo ágil es un enfoque iterativo e incremental que se realiza de manera altamente colaborativa mediante la autoorganización de equipos. Existen muchos métodos específicos de desarrollo ágil que promueven iteraciones de desarrollo, trabajo en equipo, colaboración y adaptabilidad de procesos a lo largo del ciclo de vida del proyecto. Los métodos ágiles de desarrollo frecuentemente desalientan la creación de documentación detallada de soporte de ingeniería y especificaciones técnicas detalladas. Esto significa que los desarrolladores de la información a menudo no tienen documentación fuente desde la cual extrapolar los detalles de las funcionalidades. Ahora bien, el corto período de tiempo de las iteraciones significa que la participación de cada miembro del equipo es esencial. En particular, la falta de disponibilidad del desarrollador de la información en el equipo puede hacer que sea imposible que los entregables de información se mantengan al ritmo que avance el desarrollo del software.

En este sentido, la documentación del estándar ISO/IEC/IEEE 26515 proporciona requisitos puntuales para que los roles correspondientes desarrollen información de calidad en proyectos que utilizan métodos ágiles de desarrollo considerando a dos procesos principales:

- Gestionar el desarrollo de la información, en este proceso se describen los tipos de documentación que deben desarrollarse en un entorno de desarrollo ágil. Los documentos típicos pueden incluir historias de usuarios, casos de uso, descripciones de personas, listas de tareas, informes de Scrum, y más.
- Preparar la información para los usuarios, en este proceso se establece que los requisitos de los usuarios son la base de toda la información producida en el proyecto de desarrollo ágil. Estos requisitos deberán registrarse para que los planes del proyecto y los planes de desarrollo de la información los tomen como referencia, permitiendo que las modificaciones o las características nuevas puedan rastrearse hasta los usuarios.

La Tabla 1 detalla las actividades que se realizan en cada proceso mediante una breve descripción que resume el trabajo requerido para cumplir con las especificaciones del estándar ISO/IEC/IEEE 26515.

**Tabla 1.** Procesos y actividades establecidas por el estándar ISO/IEC/IEEE 26515 (2018)

Procesos	Actividades	Descripción
Gestionar el desarrollo de la información	A1. Gestionar el cambio para el desarrollo ágil	El equipo de desarrollo de la información debe recibir capacitación sobre prácticas ágiles de desarrollo antes de que su organización comience a usar métodos ágiles. El objetivo es determinar si la información será desarrollada considerando a tales métodos o si se utilizarán otros.
	A2. Formar equipos para el desarrollo ágil	Un equipo ágil generalmente está compuesto por un arquitecto, programadores, <i>testers</i> , desarrolladores de la información, representantes de los usuarios y otras partes interesadas. Los desarrolladores de la información pueden cubrir diferentes roles y tareas.
	A3. Gestionar el desarrollo de la información en todos los equipos al utilizar el desarrollo ágil	En un proyecto de desarrollo ágil, varios equipos pueden estar trabajando sobre diferentes características del software. Por lo tanto, los desarrolladores de la información que trabajan separados pueden crear diferente información que no encaje cuando el producto se integra como un todo.
	A4. Gestionar las tareas del desarrollo de la información en las iteraciones	Durante el desarrollo ágil, la información para los usuarios se desarrolla en paralelo y dentro del mismo cronograma creado para el software por lo que se requiere su planificación, la entrega de recursos en cada iteración, la estimación del trabajo necesario, y el manejo de cambios de último momento.
	A5. Monitorear y analizar el progreso	El seguimiento del progreso es una parte importante del desarrollo ágil. Los elementos comprometidos deben completarse con éxito en la iteración y cualquier posible impedimento o retraso debe descubrirse y resolverse lo más rápido posible.
	A6. Involucrar a los <i>stakeholders</i>	Los <i>stakeholders</i> son individuos u organizaciones que tienen interés en el desarrollo del proyecto de software. La participación de éstos es importante para el desarrollo de la información puesto que brinda oportunidades para que los requisitos, prioridades, y realimentación del cliente sean obtenidos.
Preparar la información para los usuarios	A7. Establecer la relación entre el desarrollo ágil y el desarrollo de la información	En el desarrollo ágil es importante que el proceso de desarrollo de la información sea parte del ciclo de vida del producto de software y se realice junto con éste. Esto permitirá que el software y la información para los usuarios se prueben, distribuyan, y reciban mantenimiento juntos.

Procesos	Actividades	Descripción
	A8. Diseñar productos y desarrollar la información para los usuarios	Las historias de usuario, las personas, y los casos de uso deben desarrollarse en la etapa de requisitos y diseño de alto nivel del proyecto. Estos tipos de información informan a los desarrolladores de información y otros miembros del equipo de desarrollo sobre el propósito y los usos del software.
	A9. Diseñar y desarrollar la información para los usuarios	Los requisitos de un proyecto de desarrollo ágil deben formar la base de la información para los usuarios. La mayor parte de la información producida para los usuarios se asociará con el código, pero puede haber características separadas que se relacionen con los requisitos específicos del usuario que afecten solo a la información para los usuarios.
	A10. Revisar y probar la información para los usuarios	En todos los proyectos, la revisión y prueba de la información destinada a los usuarios es importante para garantizar su precisión, integridad, idoneidad, y consistencia.
	A11. Traducir y contextualizar la información para los usuarios	Puede ser que un requisito para comercializar productos en algunos países u organizaciones sea tener disponible una versión traducida o contextualizada al mismo tiempo que se realizar un lanzamiento. Por lo tanto, el equipo de desarrollo ágil debe incluir estos factores en su planificación para mantener actualizadas las versiones traducidas o contextualizadas.
	A12. Producir la información para los usuarios	Si los entregables de información para el proyecto deben someterse a un procesamiento adicional antes de entregarlos a los clientes, el tiempo requerido para esta actividad y la programación deben incluirse en la planificación del proyecto. Por ejemplo, la información impresa para los usuarios puede requerir de más tiempo para su preparación e impresión.
	A13. Liberar información para los usuarios mediante un proceso de entrega continua (DevOps)	Los equipos pueden aprovechar la integración continua y las prácticas de entrega continua para minimizar el tiempo entre la creación de un nuevo software y el envío de las nuevas funcionalidades a los usuarios o clientes. En tal escenario, a veces denominado DevOps, los equipos de desarrollo no dependen de un equipo de TI dedicado para dar servicio a su infraestructura de TI, sino que pueden atender en gran medida sus propias necesidades.

Como se puede observar, a pesar de que se trata de un resumen de los principales procesos y actividades del estándar ISO/IEC/IEEE 26515, la terminología empleada es confusa, extensa, y demasiado técnica. Por lo tanto, su enseñanza a nivel licenciatura requiere de métodos distintos al tradicional que se basa únicamente en clases teóricas. Por lo tanto, en la siguiente sección se expondrán diferentes propuestas que se basan en el uso de uno de los métodos que mejores resultados han proporcionado al área de la Ingeniería de Software: los juegos serios. En este sentido, es importante mencionar que hasta el día de hoy no existen juegos serios que aborden al estándar ISO/IEC/IEEE 26515, pero si existen propuestas que se han creado para la enseñanza de diferentes estándares creados por los mismos organismos. Con el fin de ayudar a los lectores que pudieran estar interesados sobre detalles más específicos del estándar, el documento oficial puede descargarse del siguiente enlace: [https://drive.google.com/file/d/1sD1huHBE5PsrsMTKN6IW7DP0Y-nHtC8\\_/view](https://drive.google.com/file/d/1sD1huHBE5PsrsMTKN6IW7DP0Y-nHtC8_/view).

## 2.4. Estado del arte

La Ingeniería de Software aborda el estudio y la aplicación de los principios de ingeniería al diseño, desarrollo, y mantenimiento del software con el objetivo principal de crear sistemas, servicios, o productos con calidad, considerando las limitaciones de costo y tiempo. De acuerdo con Calderón et al. (2018), para lograrlo la industria requiere del respaldo de estándares internacionales que proporcionen un conjunto de procesos, actividades, y tareas que cubran el ciclo de vida del software con el objetivo de resolver correctamente los problemas que surjan durante el desarrollo del mismo. En este sentido, el acelerado desarrollo de las TI y la comunicación han posicionado a la Ingeniería de Software como una de las disciplinas más requeridas en el mercado laboral, lo que ha generado presión en las instituciones educativas para que éstas incorporen enfoques más innovadores que mejoren su enseñanza (Anicic y Stapic, 2022), actualicen constantemente los planes de estudio para los diversos cursos, y adopten estrategias de enseñanza/aprendizaje más atractivas para los estudiantes (Yasin et al., 2021). Sin embargo, a pesar de la importancia que la Ingeniería de Software ha cobrado en la actualidad, es una dura realidad que las universidades siempre han tenido restricciones financieras que han dificultado a sus autoridades la toma de decisiones importantes sobre la educación. Una de las consecuencias más importantes de esta situación es que desde la década de los 90's ha existido una diferencia importante entre las habilidades duras y blandas que los egresados desarrollan dentro de las instituciones educativas y las requeridas por la industria, situación que afecta negativamente a la industria de software. Cico et al. (2021) argumentaron, por ejemplo, que la educación sobre “procesos de desarrollo de software” y “estándares del proceso de software”, tópicos importantes dentro de las áreas temáticas de la Ingeniería de Software, requerían de un enfoque de enseñanza/aprendizaje más práctico y realista de los habituales.

Como consecuencia de lo anterior, durante los últimos años se ha observado un interés renovado por desarrollar y usar nuevos métodos y técnicas para enseñar la Ingeniería de Software de una forma más práctica de tal manera que se promueva el aprendizaje activo e interactivo, aumente la motivación y el compromiso de los estudiantes, y se diseñen nuevas estrategias para la formación y capacitación de los futuros ingenieros (Anicic y Stapic, 2022). Tales estrategias incluyen al aprendizaje basado en problemas, aprendizaje colaborativo, aprendizaje basado en proyectos, aprender haciendo (“*learning by doing*”), aprendizaje basado en juegos, entre muchos otros. El aprendizaje basado en juegos, que involucra precisamente al concepto de “juego serio”, frecuentemente es percibido de forma errónea como el simple uso de un juego en el aula, no obstante dicho enfoque implica el introducir material, estrategias, y actividades educativas a un juego con el objetivo de proporcionarle a los estudiantes un entorno donde apliquen sus conocimientos y habilidades de manera más dinámica y real, de tal forma que se incremente su interés por aprender tópicos que a menudo requieren de mucha práctica (Plass et al., 2015). Pieper et al. (2017) y Calderón et al. (2018b) argumentaron que los juegos serios se estaban convirtiendo en un campo de oportunidad que crecería rápidamente puesto que atraería la atención de profesionales e investigadores como un enfoque alternativo a los enfoques pedagógicos tradicionales puesto que un juego puede ayudar a los estudiantes a descubrir y experimentar conocimientos mediante el uso de historias, mecánicas, componentes, y dinámicas. En la actualidad se puede encontrar evidencia en la literatura especializada que demuestra la capacidad de los juegos serios para establecer un proceso de enseñanza/aprendizaje más práctico y realista que permite que los estudiantes adquieran experiencia real y al mismo tiempo se involucren en el aprendizaje de la Ingeniería de Software desde una perspectiva social, divertida, y efectiva. A continuación, se describen cinco juegos serios que fueron creados para mejorar el entrenamiento de estudiantes de nivel licenciatura en estándares del proceso de software, objetivo que persigue esta tesis con el estándar ISO/IEC/IEEE 26515.

## **2.4.1. Enseñando el estándar ISO/IEC 12207:1995 *Systems and software engineering – Software lifecycle processes*: Un enfoque basado en un juego serio**

### **2.4.1.1. Objetivo**

“Floors” es un juego serio que propuso una experiencia interactiva de aprendizaje para inculcar en estudiantes de Ingeniería en Computación los fundamentos del estándar ISO/IEC 12207:1995 *Systems and software engineering – Software lifecycle processes*, el cual es un estándar internacional que describe los procesos del ciclo de vida del software y que, años después, llegaría a ser el estándar de facto para definir todas las actividades requeridas para desarrollar y mantener el software.

### **2.4.1.2. Descripción**

Aydan et al., (2017) crearon “Floors” como un juego serio que simula nueve diferentes pisos de una oficina virtual simbolizando las partes del estándar ISO/IEC 12207:1995. Se trata de un juego digital que fue desarrollado con el lenguaje de programación C# en el entorno para videojuegos *Unity3D* con la ayuda de *Unity* y *Blender* para la creación de objetos 3D, *Mixamo Fuse* para la creación de animaciones y personajes, además de varios objetos de oficina como computadoras, impresoras, sillas, sofás, escritorios, y equipos de iluminación que fueron descargados desde la *Unity Asset Store*. El juego contiene *scripts* definidos específicamente para promover la interacción con el usuario y 22 personajes no jugables (NPC, por sus siglas en inglés), de los cuales 14 se utilizan activamente durante el juego como responsables de cada proceso seleccionado del estándar. De esta manera, el juego explica las definiciones y terminología empleadas por el estándar, pero en forma de diálogos que presentan dicha información al jugador a lo largo del juego de una manera más explicativa. Los otros 8 NPC no tiene interacción directa con el jugador, pero contienen *gifs* animados que reflejan la atmósfera del piso especificado. La Figura 2.1, por ejemplo, muestra una captura de pantalla del diálogo que se da entre el jugador y el NPC responsable del piso relacionado con los procesos de verificación y validación.

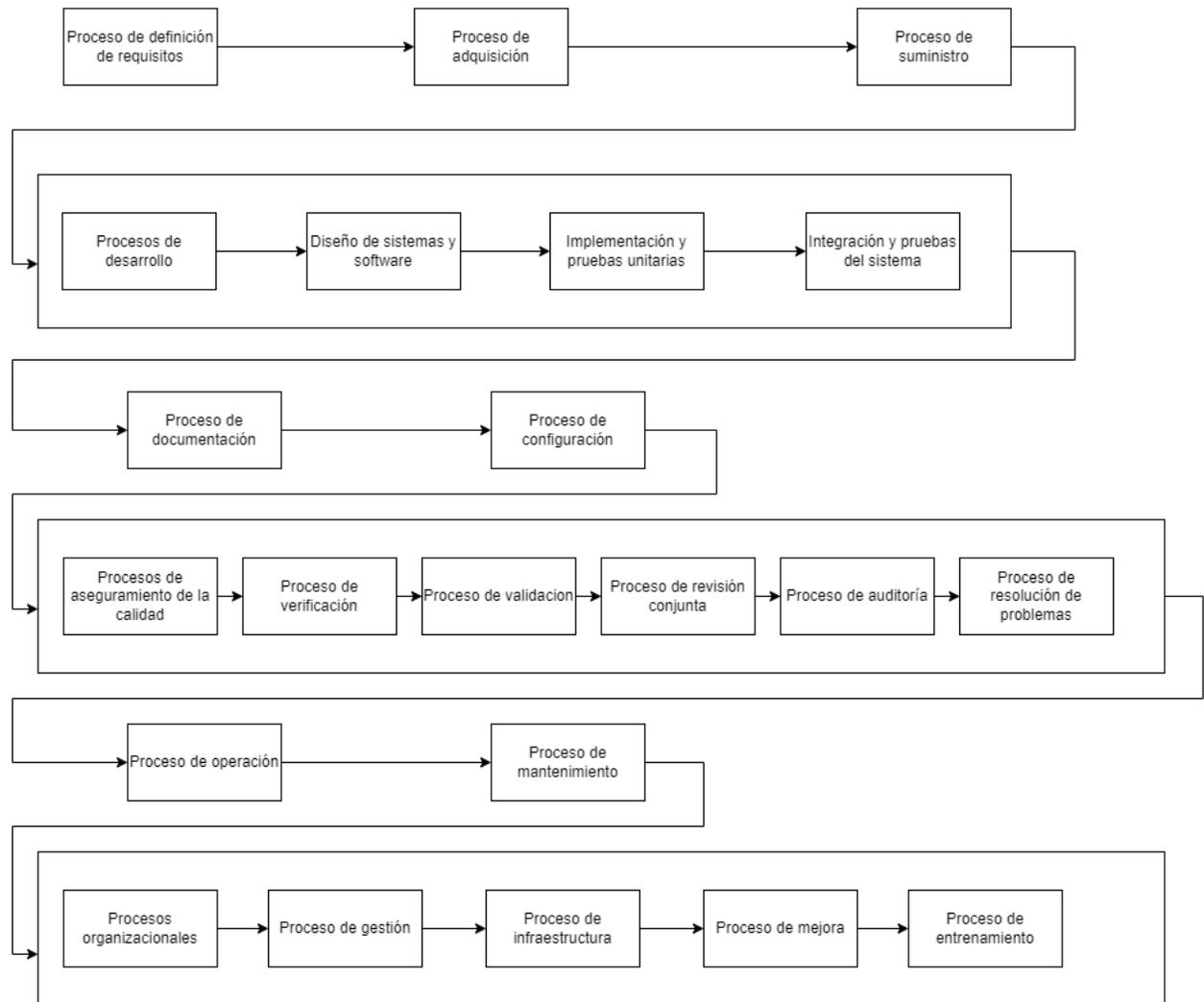
Como se observa en dicha figura, se desarrolló un entorno de oficina virtual de una organización donde se realiza la gestión de proyectos de software siguiendo el proceso descrito en el estándar ISO/IEC 12207:1995. Dicho estándar puede ser utilizado con diferentes modelos de ciclo de vida. Sin embargo, la organización debe elegir tanto los modelos como los procesos del estándar que serán utilizados antes del inicio del desarrollo de cada proyecto. Considerando esta premisa, “Floors” crea un entorno virtual para adaptar y realizar los procesos definidos por el estándar siguiendo el modelo en cascada y proporcionando información útil durante la sesión de juego. De hecho, el juego puede utilizar los procesos principales y más utilizados del estándar con especial orientación a los modelos de desarrollo tradicionales que son cascada, incremental, e iterativo. De esta manera, el jugador debe seguir una ruta definida entre los procesos del estándar y las explicaciones que le proporcionan los personajes para completar diferentes pisos. Para lograr este fin, las definiciones que proporciona la documentación oficial del estándar fueron codificadas en C# a través de un formato adecuado para definir tanto la estructura de datos de la matriz correspondiente al entorno del juego como las condicionantes anidadas que corresponden con la secuencia de los diálogos. Por lo tanto, cada NPC tiene su propia clase que contiene información sobre un proceso específico del estándar y sus actividades relacionadas.



**Figura 2.1.** Diálogo entre jugador y NPC responsable del piso relacionado con los procesos de verificación y validación de software (tomada de Aydan et al., 2017)

“Floors” proporciona a los jugadores una visión del mundo virtual parecida a su propia percepción de la vida real. De esta forma se plantea que los estudiantes se sientan más inmersos durante la sesión de juego. En este sentido, para crear un entorno de juego más receptivo y atraer la atención de los jugadores se creó un conjunto de eventos similares a la vida real que siguen un proceso iterativo basado en diálogos técnicos y lecturas que se desarrollan en oficinas simuladas en 3D. En el juego se pueden visualizar los escenarios organizacionales de la vida real con la ayuda de un entorno de oficina virtual y modelos de personajes representativos que explican las definiciones y actividades de los procesos del estándar. De esta manera, un jugador realiza búsquedas y participa en diálogos específicos mientras sigue los procesos del estándar en un entorno de oficina virtual. La Figura 2.2 muestra el diagrama de los procesos del estándar referentes al modelo de ciclo en cascada que un jugador debe seleccionar y su secuencia de ocurrencia durante el juego, es decir una cadena de tareas. Para que los jugadores estén activos durante el juego, se añadieron varias misiones en diferentes pisos mientras se relacionan con los procesos definidos por el estándar. De esta manera, los jugadores se involucran directamente con los procesos y las actividades que éstos requieren para su implementación.

Para determinar si el juego cumplía con su función educadora o no, se condujo un estudio que evaluó la experiencia de los estudiantes durante las sesiones de juego. Dichos estudiantes recibieron inicialmente clases teóricas sobre el estándar ISO/IEC 12207:1995 siguiendo un enfoque tradicional de clases en el aula y posteriormente fueron separados en dos grupos con el objetivo de realizar la parte práctica del curso. Uno de los grupos realizó la práctica habitual utilizando la documentación oficial del estándar, mientras que el otro utilizó el juego “Floors” para realizar la misma tarea. Para medir las experiencias de los jugadores se consideraron las recomendaciones de Bernhaupt (2011) al aplicar encuestas y realizar entrevistas que adoptaron una versión de *Game Play* (Parnell et al., 2009), que utiliza una escala Likert para evaluar y medir la experiencia de los jugadores durante las sesiones de juego. Posteriormente, al finalizar las sesiones de juego, un cuestionario que contenía una combinación de preguntas sobre el estándar, la gestión de proyectos, y conceptos básicos de Ingeniería de Software fue aplicado a ambos grupos de estudiantes con el objetivo de determinar su nivel de conocimientos y el desempeño de cada grupo.



**Figura 2.2.** Diagrama de procesos seleccionados del estándar ISO/IEC 12207:1995 y su secuencia de ocurrencia durante el juego (traducida de Aydan et al., 2017)

### 2.4.1.3. Resultados

40 estudiantes con una edad promedio de 21 años participaron en la evaluación empírica de “Floors”, 19 fueron mujeres y 21 hombres. Estos estudiantes cursaban diferentes años de los programas académicos de la Licenciatura en Informática e Ingeniería en Computación. Los datos recogidos mostraron que el grupo de estudiantes que utilizó el juego serio para realizar la parte práctica del curso obtuvo un aumento en sus habilidades de comprensión y aplicación de los procesos del estándar ISO/IEC 12207:1995. Debido a que se realizaron diferentes sesiones de juego, se utilizó la prueba *t-test* para dos muestras independientes de estudiantes con el objetivo de realizar un análisis de los datos obtenidos en la evaluación empírica mediante el uso de IBM *SPSS Statistics 23*. Para realizar la prueba se consideró un nivel de significancia del 95% (0.05). Se utilizó un valor del grado de libertad ( $df = N - 2$ ) de 38 puesto que la muestra de participantes fue de 40 estudiantes ( $N = 40$ ). Todos los cálculos independientes de la prueba *t-test* se obtuvieron a través de la computadora, por lo que la probabilidad fue demasiado pequeña para ser considerada. De acuerdo con estos valores, el valor *t* se calculó como 4.27 ( $t = 4.27$ ). Estos resultados condujeron a los investigadores a determinar

que los estudiantes que participaron en las sesiones de aprendizaje con el juego serio tendieron a experimentar situaciones más positivas, mientras que los estudiantes que participaron en las sesiones de aprendizaje tradicional reportaron experiencias ligeramente neutrales de acuerdo con las puntuaciones del cuestionario con la escala *Gameplay*. Con el objetivo de determinar la exactitud de la variabilidad se consideró a la  $d$  de *Cohen*, que es una medida para encontrar la dimensión del efecto, en este caso el efecto que el juego serio tuvo sobre los estudiantes. La fórmula de dicha medida utiliza  $S_1$  y  $S_2$  para representar las desviaciones estándar de las muestras independientes, cuyos valores fueron 13.20 y 3.49 respectivamente. De acuerdo con estos valores, la varianza combinada estándar se calculó como 9.65 ( $S_p = 9.65$ ) y la  $d$  de *Cohen* se calculó como 1.35 ( $d = 1.35$ ). Por lo tanto, para indicar qué tan efectivo fue este resultado, se utilizaron los valores definidos para  $d$ , que son 0.2 (efecto pequeño), 0.5 (efecto mediano) y 0.8 (efecto grande). Dado que la dimensión del efecto de este estudio fue grande ( $d = 1.35$ ), el resultado fue 0.32. De acuerdo con la escala que se define para los valores de  $r^2$ , es decir 0.01 (efecto pequeño), 0.09 (efecto mediano), y 0.25 (efecto grande), el juego serio tuvo un efecto grande en los estudiantes puesto que  $r^2 = 0.32$  (32.0%). Además, la evaluación de los conocimientos que los estudiantes adquirieron a través de las sesiones de juego confirmó que éste puede ser utilizado como una herramienta educativa para entrenar a estudiantes de nivel licenciatura en el estándar. Se concluyó que los juegos serios no solamente podrían ofrecer mejoras sustanciales y duraderas en la enseñanza de los estándares del proceso de software, sino que también podrían tener implicaciones importantes en la educación en general.

## **2.4.2. Un juego serio para apoyar la enseñanza del estándar ISO 21500:2012 en el contexto de la gestión de proyectos de software**

### **2.4.2.1. Objetivo**

“ProDec” fue desarrollado como un recurso de aprendizaje para que los estudiantes o profesionales de la Ingeniería de Software pongan en práctica sus conocimientos relacionados con los procesos de gestión del estándar ISO 21500:2012 en los proyectos de software a través de un entorno de aprendizaje/enseñanza que promueve el aprendizaje práctico, activo, y social.

### **2.4.2.2. Descripción**

Calderón et al. (2018a) crearon “ProDec” como un juego serio de simulación que tiene como propósito capacitar y apoyar a los estudiantes en el entendimiento de los procesos de gestión de proyectos de acuerdo con el estándar ISO 21500:2012 *Guidance on project management*. De acuerdo con sus creadores, el juego tiene cuatro principales funcionalidades: (a) proporciona un entorno de entrenamiento que permite a los estudiantes acercarse a todas las etapas del ciclo de vida de un proyecto de software, desde su concepción hasta su cierre; (b) proporciona a los profesores un entorno para apoyar la evaluación de los estudiantes a través de sesiones de juego; (c) proporciona tanto a los estudiantes como a los profesores un entorno para el diseño de escenarios de juegos que les permite crear todos los escenarios de planes de proyectos que se les ocurran; y (d) proporciona un entorno de aprendizaje/enseñanza que promueve el aprendizaje práctico, y el aprendizaje activo y social.

La dinámica principal del juego promovió que los estudiantes asumieran el rol de jefes de proyectos en un entorno virtual de tal forma que pudieran gestionar exitosamente un proyecto de software, lo que implicó que aprendieran a hacerlo dentro de los límites establecidos de tiempo y costo. El juego termina cuando se ha completado el proyecto de forma exitosa o cuando se hayan superado significativamente el presupuesto aprobado o el tiempo asignado. La Figura 2.3 muestra que los estudiantes reciben una declaración que define el alcance del proyecto involucrado en el escenario

del juego para posteriormente iniciar el proceso del ciclo de vida del juego con el objetivo de ganar la partida. Como resultado, los estudiantes terminan el juego obteniendo un informe de evaluación que les permite aprender de su propio desempeño al jugar. El proceso del ciclo de vida del juego se compone de tres etapas principales: inicio, ejecución, y cierre.

**Figura 2.3.** Representación del ciclo de vida de “ProDec” (traducida de Calderón et al., 2018a)

En la *etapa de inicio* los jugadores siguen un proceso que los orienta en la creación, desde cero, del escenario de un juego, el cual involucra la definición del plan de proyecto que reflejará la manera específica en que los jugadores interactuarán con el juego. A su vez, esta etapa contiene cinco subetapas secuenciales que se definen de la siguiente manera:

- Documentación de la información del proyecto: los jugadores proporcionarán la información general del proyecto respecto a su alcance y características para comenzar la estimación de su tamaño (e.g., duración, cantidad de casos de uso, salario de los trabajadores, etc.).
- Estimación del tamaño: los jugadores deberán hacer la estimación del tamaño del proyecto planteado mediante el método de estimación base de puntos de función de Albrecht (1979).

- Definición del equipo del proyecto: los jugadores deberán definir su equipo de proyecto. Para lograrlo deberán considerar su experiencia anterior y algunos rasgos de personalidad en base a los dieciséis factores descritos por Cattell et al. (1970).
- Definición de tareas: los jugadores deberán definir las tareas del proyecto utilizando un diagrama PERT (Moder et al., 1983), además registrarán la duración, presupuesto asignado, tareas predecesoras, y personal requerido para cada una de éstas.
- Análisis de riesgos: los jugadores realizarán un análisis cuantitativo de riesgo.

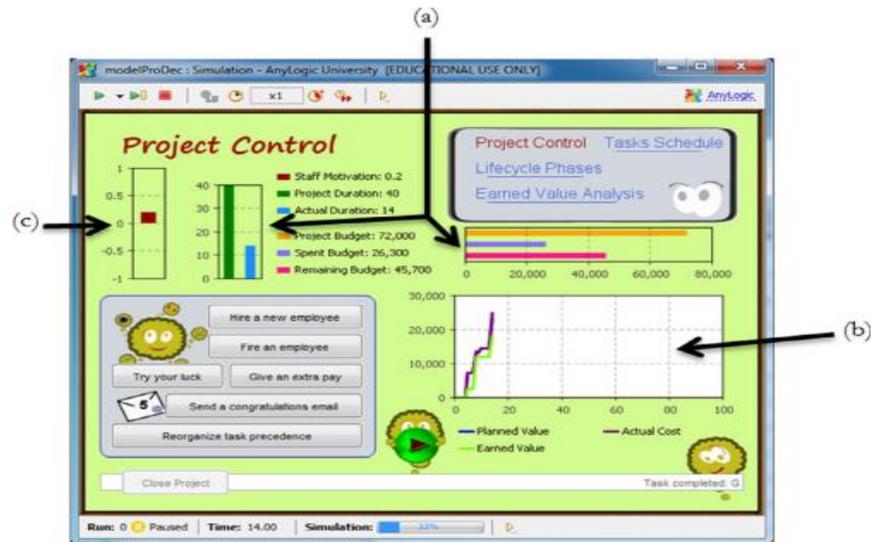
Una vez que el plan del proyecto se haya definido, se generará automáticamente un archivo de código fuente que contendrá ecuaciones de un modelo de simulación de eventos discretos que corresponde con la ejecución de dicho plan, permitiendo así que los jugadores inicien la etapa de ejecución. En este punto, “ProDec” utilizará la información del plan del proyecto para generar automáticamente el código de un modelo con *AnyLogic*, la cual es una herramienta de simulación multimétodo que permite crear modelos de simulación modulares y jerárquicos que se pueden obtener siguiendo metodologías de simulación basada en agentes, dinámicas, y/o eventos discretos. El código generado por “ProDec” es, como se mencionó anteriormente, un modelo de simulación de eventos discretos que describe la colección de clases y objetos para definir el flujo de trabajo del proyecto seleccionado, así como los recursos asociados, riesgos y limitaciones representados en el plan del proyecto. Además, también se crea automáticamente el código necesario para mostrar y administrar una interfaz de usuario ad-hoc para el modelo de simulación que se genera. Dicha interfaz permite a los jugadores interactuar con el modelo de simulación tomando diversas decisiones respecto a las tareas y los recursos que se asignaron en la etapa anterior, facilitando así la visión del efecto que ambos tienen en la evolución del proyecto. De acuerdo con sus creadores, esta característica le da a “ProDec” la flexibilidad suficiente para simular cualquier plan de proyecto de software que el jugador desee gestionar. Una vez construido, el modelo de simulación se ejecuta para constituir el juego ejecutable y los jugadores comienzan a gestionar el proyecto. Durante la simulación de la ejecución del plan del proyecto, el juego permite que los jugadores practiquen sus habilidades en la toma de decisiones mediante el control y monitorización del progreso del proyecto con el objetivo de corregir las desviaciones potenciales en el mismo. La Figura 2.4 muestra una pantalla de control a través de la cual se visualiza el progreso del proyecto en tiempo real mediante los siguientes elementos: (a) el tiempo y presupuesto gastados y restantes, (b) los resultados del análisis del valor ganado sobre el progreso del proyecto, y (c) el nivel de motivación del equipo del proyecto.

Cuando la simulación del plan de proyecto haya finalizado, los jugadores avanzan a la etapa de cierre del proceso de ciclo de vida del juego. En este sentido, “ProDec” utiliza la información que se ha generado durante la sesión de juego y los criterios de evaluación preestablecidos por los profesores para generar un reporte de evaluación con toda la información detallada sobre todos los eventos ocurridos en el juego, los cuales describen las habilidades adquiridas por los estudiantes. Dicho reporte permite a los estudiantes y profesores estudiar el curso del juego, facilitando así el análisis de las decisiones tomadas y los resultados obtenidos. En resumen, en esta última etapa se realiza el cierre del proyecto y se obtiene un reporte de evaluación del desempeño de los estudiantes durante la sesión de juego.

### 2.4.2.3. Resultados

Con el objetivo de evaluar la eficacia de “ProDec” como herramienta de soporte a la enseñanza del estándar ISO 21500:2012, se realizó una evaluación empírica dentro del curso 2016/2017 de Desarrollo Global de Software (GDS, por sus siglas en inglés) en la Universidad de Cádiz, España, la cual contó con la participación de 24 estudiantes de nivel licenciatura de la carrera de Ingeniería de

Software. El proceso de evaluación consideró al cuestionario propuesto por Savi et al. (2011) en el Modelo para la Evaluación de Juegos Educativos para enseñar la Ingeniería de Software (MEEGA, por sus siglas en inglés) con el fin de determinar la eficacia educativa del juego en términos de motivación, experiencia del usuario, y resultados de aprendizaje. La aplicación de este cuestionario se realizó una vez que las sesiones de juego habían terminado. Además, con el objetivo de valorar las percepciones de los estudiantes con respecto a los objetivos de aprendizaje establecidos por el juego, se agregaron preguntas al cuestionario para evaluar los tres niveles establecidos por la Taxonomía de Bloom (Anderson y Krathwohl, 2001), es decir conocimiento, comprensión, y aplicación.



**Figura 2.4.** Pantalla de ejemplo de la etapa de ejecución de “ProDec” (tomada de Calderón et al., 2018a)

Una vez que se recogió y analizó la información de los cuestionarios se llegó a las siguientes conclusiones. Todos los ítems del cuestionario alcanzaron un valor mediano positivo de al menos un valor de 1 dentro del rango  $[-2, +2]$ , con la excepción del ítem relacionado con la dimensión “diversión” de la categoría “Experiencia de usuario”, cuya media alcanzó un valor 0 (i.e., Ni en desacuerdo ni de acuerdo). Aunado a esto, los estudiantes consideraron que “ProDec” les brindó un buen nivel de “inmersión”, “relevancia”, y “desafío” como recurso de aprendizaje. Por lo tanto, considerando la información recogida, se concluyó que los estudiantes evaluaron positivamente los *ítems* sobre motivación, experiencia de usuario, y aprendizaje luego de su experiencia práctica con el juego. Por otro lado, con relación a la realimentación proporcionada por los estudiantes en lo referente al aprendizaje percibido, el 75% de los estudiantes consideró que “ProDec” les ayudó a recordar, comprender, y aplicar los grupos temáticos de Tiempo, Costo, Riesgos, Integración, Alcance, Recursos, y *Stakeholders* que son definidos por el estándar ISO 21500:2012. Además, los estudiantes también coincidieron en que el juego les ofreció una ayuda importante para recordar las actividades del grupo temático de *Stakeholders*, las cuales fueron las más difíciles de comprender en la práctica. En contraste, cuatro estudiantes consideraron que el juego no contribuyó a mejorar la comprensión del grupo temático de Riesgo, siendo el único *ítem* que alcanzó un valor bajo de contribución.

Finalmente, las opiniones de los estudiantes relacionadas con las fortalezas y debilidades del juego relevaron que: “ProDec” es fácil de usar como herramienta de aprendizaje, la experiencia con el juego es divertida, “ProDec” sirve como ayuda para poner en práctica la mayoría de los procesos definidos en el estándar ISO 21500:2012, “ProDec” permite explorar los cinco grupos de procesos de dicho estándar (i.e., Inicio, Planificación, Implementación, Control, y Cierre), la interfaz gráfica de

usuario del juego debe ser mejorada, y es necesario agregar más características al juego. Sin embargo, se observó que existía una escases de trabajos similares que ayudaran a los estudiantes a comprender y a los profesores a enseñar el estándar ISO 21500:2012 mediante enfoques más prácticos y realistas que permitan poner en práctica los conocimientos adquiridos en escenarios simulados de la vida real.

### **2.4.3. ProcSoft: Un juego de tablero para enseñar los procesos de software en base al estándar ISO/IEC 29110**

#### **2.4.3.1. Objetivo**

“ProcSoft” es un juego no digital que se creó con el objetivo de promover entre estudiantes de nivel licenciatura la correcta definición de un proceso de software de acuerdo con el estándar ISO/IEC 29110:2018 *Systems and software engineering – Lifecycle profiles for very small entities*. En este sentido, dicho juego serio establece actividades complementarias para facilitar la asimilación del contenido mediante la definición de actividades que son comunes en un ciclo de vida del software, el establecimiento de su relación con los roles existentes, y la familiarización con herramientas de soporte que pueden usarse para mejorar su desempeño. Así mismo, el juego pretendió que los estudiantes comprendieran la importancia de asignar actividades correctas a un rol específico, la necesidad de utilizar las herramientas apropiadas para realizar las actividades, y la importancia de utilizar la Ingeniería de Software en la ejecución de un proceso.

#### **2.4.3.2. Descripción**

“ProcSoft” (Moura y Santos, 2018) es un juego de mesa que simula el escenario donde un equipo de trabajo debe crear un proceso para el desarrollo de un software específico. Para lograr este objetivo, el juego dispone de un tablero, una plantilla para comprobar el proceso creado (i.e., identifica las fases a las que se refieren las actividades y los roles asociados en un flujo de cascada), un cuadernillo de reglas, tarjetas de actividades, tarjetas de roles y herramientas, tarjetas con preguntas sobre la Ingeniería de Software, tarjetas de suerte/castigo (i.e., beneficios y perjuicios para el proyecto), peones, dados, y billetes (dinero). Los estudiantes deben formar dos equipos de 2 a 4 integrantes que interactuarán con un solo tablero para crear el proceso más adecuado de software considerando restricciones de recursos financieros, material, y recursos humanos.

El tablero refleja las actividades más relevantes del estándar ISO/IEC 29110:2018, las cuales son sintetizadas y distribuidas a lo largo de 44 casillas que definen la naturaleza del juego. En dicho tablero también se incluyeron cuatro casillas especiales: suerte/castigo, \$\$ (que se refiere al dinero), rol, y herramientas, las cuales tienen la intención de agregar dinamismo al juego y presentar facilidades o dificultades a los jugadores. Además, estas casillas intentan representar diversas necesidades y eventos que pueden ocurrir durante la creación y ejecución de los procesos de software en una organización. La Figura 2.5 muestra que el tablero se divide en cuatro grupos y colores de acuerdo con los roles definidos para el juego: jefe de proyectos – morado, analista – rosa, programador – verde, y *tester* – naranja. Por lo tanto, cada rol simula las responsabilidades que existen en los equipos reales de desarrollo de software. Los últimos tres roles tienen dos niveles de experiencia, *junior* y *senior*.

Por otro lado, las tarjetas de actividades representan tareas de las cuatro etapas definidas por el estándar ISO/IEC 29110:2018: planificación, ejecución y seguimiento, análisis y diseño, y codificación/pruebas/entrega. Para adquirirlas durante una sesión de juego, es necesario que un equipo se detenga en la casilla correcta y responda una pregunta relacionada con la Ingeniería de Software.

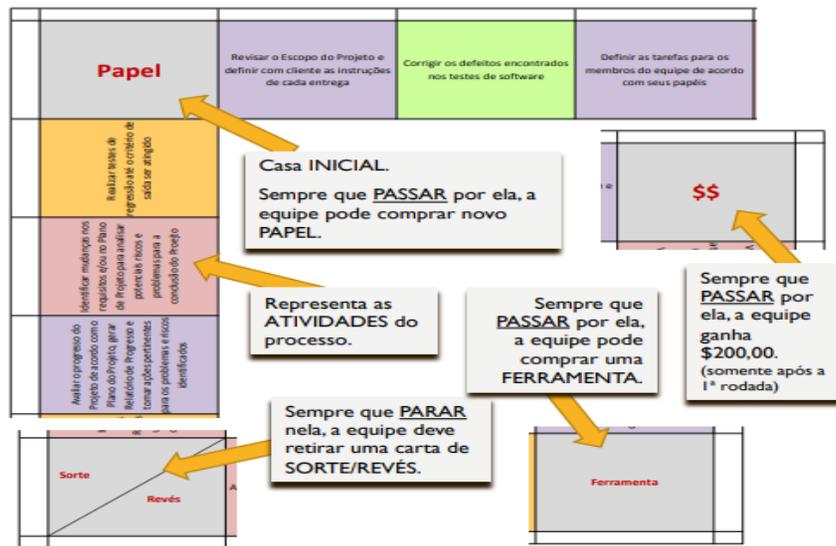


Figura 2.5. Tablero y casillas especiales del juego “ProcSoft” (tomada de Moura y Santos, 2018)

De manera similar, el juego introduce la posibilidad de que los jugadores hagan uso de herramientas que los ayuden a realizar las tareas anteriormente mencionadas en gestión, análisis y diseño, codificación y pruebas. Algunas herramientas solamente pueden ser utilizadas por ciertos roles y niveles, además de que su precio varía de acuerdo con su calidad/posición en el mercado. Para determinar el costo de cada actividad y herramienta se utiliza la información mostrada en la Tabla 2.

Tabla 2. Conteo de puntos finales del proceso creado (traducido de Moura y Santos, 2018)

Tarjeta	Situación	Costo
Actividad	Con la herramienta adecuada	\$100
Actividad	Sin la herramienta adecuada	\$50
Actividad	Si se tiene un nivel <i>senior</i> para realizarla	+ \$50
Herramienta	--	Valor nominal
Herramienta	Si se asocia con un nivel <i>senior</i>	+ \$50 por actividad

Antes de iniciar la sesión de juego, el dinero se distribuye entre los equipos. Cada equipo recibe \$2,100 dividido en dos billetes de \$500, cinco billetes de \$100, y 12 billetes de \$50. El juego inicia en la casilla de rol y un jugador puede promocionar a uno de sus roles (i.e., de junior a senior) si al tirar los dados el número de movimientos resultante lo hacen llegar a ella dentro del tablero. La promoción se puede realizar siempre y cuando se tenga dinero suficiente para pagar la mejora y el jugador tendría acceso a utilizar más herramientas y recibir ciertas bonificaciones. En la primera ronda los equipos pueden comprar dos roles, los cuales definirán un criterio de habilidad para avanzar en el tablero. Solamente será posible comprar una actividad cuando el jugador llegue a la casilla correspondiente y responda una pregunta vinculada a un rol específico. La pregunta será formulada por el otro equipo, siempre y cuando el jugador posea la tarjeta del rol correspondiente. Cada actividad tiene un costo inicial de \$100, el cual incrementará en \$100 si el equipo responde mal a la pregunta o en \$50 si no cuenta con la herramienta adecuada. Así mismo, solo se podrán manipular herramientas si el equipo tiene un rol del nivel adecuado. Cada equipo realizará un lanzamiento de dados en su turno para avanzar en el sentido de las agujas del reloj. Si el lanzamiento produjera un doblete, el equipo tendrá derecho a un nuevo lanzamiento. Cuando un jugador cae en la casilla de una actividad

que ya fue comprada anteriormente por el equipo contrario, deberá robar una carta de suerte/castigo. Si la tarjeta es de suerte, el equipo propietario de la actividad recibirá una bonificación, pero en cambio, si la tarjeta es de castigo, el equipo que es propietario de la actividad deberá hacerle un pago al jugador.

Ahora bien, si el equipo no tiene dinero para pagar, sus propiedades pueden ser vendidas siguiendo el orden de negociación: devolver herramientas, bajar el nivel de algún rol, o devolver tarjetas de actividades. Si aun después de la negociación no puede pagar sus deudas, el equipo se declarará en bancarrota y se retirará del juego. Durante el juego ningún jugador podrá regalar o prestar dinero. La duración sugerida para las sesiones de juego es de 60 a 90 minutos; no obstante, el juego puede terminar una vez que se alcance el tiempo acordado entre los equipos. El ganador será quien logre comprar la mayor cantidad de actividades, haciendo el mejor uso de los roles y las herramientas para obtener una mayor bonificación.

### 2.4.3.3. Resultados

La evaluación empírica de “ProcSoft” también hizo uso de MEEGA (Savi et al., 2011), el cual define 29 ítems estandarizados divididos en tres subcomponentes (i.e., motivación, experiencia del usuario en el juego, y aprendizaje) que evalúan la reacción de los estudiantes ante el juego, además de que se incluyeron preguntas personalizadas sobre los contenidos tratados. Cada subcomponente se divide en dimensiones y éstas a su vez están asociadas a diferentes preguntas. Cada pregunta se responde utilizando una escala Likert de 5 valores posibles: “Totalmente de acuerdo”, “Estoy de acuerdo”, “No estoy de acuerdo ni en desacuerdo”, “No estoy de acuerdo”, y “Estoy totalmente en desacuerdo”. La evaluación consideró el uso de un cuestionario electrónico (*Google Forms*) que estuvo disponible después de que los estudiantes utilizarán el juego en el aula. En esta evaluación participaron 12 estudiantes de la Universidad Federal del Estado de Río de Janeiro, de los cuales 10 respondieron el cuestionario utilizado. Se formaron cuatro equipos de tres participantes cada uno. Los equipos se formaron inicialmente con estudiantes presentes al comienzo de la clase y algunos miembros se agregaron más tarde. La sesión de juego tomó aproximadamente 90 minutos de los 120 disponibles para la clase.

La motivación se evaluó considerando cuatro dimensiones: atención, relevancia, confianza, y satisfacción. En cuanto a la atención, el diseño del juego fue considerado atractivo por el 50% de los estudiantes, mientras que el 30% de éstos reportó lo contrario y se obtuvieron dos respuestas neutrales. El 100% de los estudiantes declaró que el juego captó su atención desde el principio; sin embargo, el 90% de éstos consideró que las actividades que debían realizar terminaron siendo monótonas y aburridas. Respecto a la relevancia, la evaluación fue positiva puesto que el 80% de los estudiantes estuvo de acuerdo en que el juego era importante, el 100% afirmó que el funcionamiento del juego era el adecuado para aprender, y la mayoría (70%) consideró que el juego se relacionaba con el conocimiento adquirido en el aula (con 3 participantes neutrales). Con relación a la confianza, el número de participantes que reportaron acuerdo y desacuerdo en cuanto a la facilidad de comprensión de las reglas fue el mismo (4 en cada grupo). No obstante, se registró un 90% de respuestas positivas que confirmaron que el juego fomentó el aprendizaje a lo largo de su uso. Por último, los estudiantes demostraron niveles moderados de satisfacción al coincidir en que el contenido abordado por el juego podría ser utilizado en la práctica (60% de respuestas positivas y 20% de negativas) y que el esfuerzo personal determinó el progreso en las tareas realizadas (50% positivas y 30% negativas).

Por otro lado, la experiencia de usuario se evaluó mediante cinco dimensiones: competencia, diversión, desafío, interacción social, e inmersión. Para la competencia se obtuvo un alto grado de acuerdo entre los estudiantes, puesto que el 80% afirmó que fue capaz de lograr los objetivos a través

de sus propias habilidades y sentirse eficientes durante la sesión de juego. No se observaron respuestas negativas. En cuanto a la diversión, el 70 % de los estudiantes informó que volvería a jugar y recomendaría el juego a sus colegas. Tampoco se obtuvieron respuestas negativas. El juego demostró ser divertido para el 100% de los participantes. Sin embargo, el 20% (2 de 10) informó no sentirse decepcionados cuando terminó el juego, mientras que el 40% dijo lo contrario. Con respecto al desafío, todos los estudiantes coincidieron en que el juego fue desafiante, el 70% reportó que el juego no fue monótono y estableció un ritmo adecuado; sin embargo, el 20% no estuvo de acuerdo. Con relación a la interacción social, todos los estudiantes argumentaron que el juego promovió la cooperación, la competencia y la interacción, además de que les brindó diversión. Desafortunadamente, la evaluación de la inmersión presentó la mayor cantidad de respuestas negativas puesto que solamente la mitad de los estudiantes confirmó que el juego fue capaz de desconectarlos completamente del mundo real. Aunque el 70% de los estudiantes informó que se distanció de las preocupaciones cotidianas durante el juego, el 20 % de los participantes informó lo contrario. Sin embargo, a pesar de la cantidad de respuestas discordantes, la evaluación general de la inmersión para la mayoría de los participantes se consideró positiva.

Finalmente, la evaluación del aprendizaje a corto plazo devolvió resultados positivos puesto que el 90% de los estudiantes reconoció la eficacia del juego para mejorar el aprendizaje y su aporte a la disciplina de la Ingeniería de Software.

#### **2.4.4. Un juego serio para enseñar los conceptos fundamentales de la Ingeniería de Requisitos y del estándar ISO/IEC/IEEE 29148:2011 a nivel licenciatura**

##### **2.4.4.1. Objetivo**

“Requenguin” fue desarrollado con el objetivo principal de proporcionar a los estudiantes universitarios un entorno de aprendizaje interactivo que facilite la comprensión y aplicación de los procesos principales del estándar ISO/IEC/IEEE 29148:2011, así como el aprendizaje de algunas técnicas relacionadas con la Ingeniería de Requisitos.

##### **2.4.4.2. Descripción**

“Requenguin” (García et al., 2020) simula una biblioteca universitaria en 3D donde los jugadores deben aplicar los procesos de la Ingeniería de Requisitos con el objetivo de cambiar el sistema tradicional de gestión por un sistema basado en software mientras reciben formación preliminar en el estándar ISO/IEC/IEEE 29148:2011 *Systems and software engineering — Life cycle processes — Requirements engineering* y, al mismo tiempo, mejoran algunas habilidades técnicas en este campo (véase Figura 2.6). El juego pretende que los estudiantes mejoren su comprensión de los dos procesos definidos por dicho estándar: definición de los requisitos de los *stakeholders* y análisis de los requisitos, mientras exploran una biblioteca universitaria con el objetivo de obtener un conjunto de requisitos de software necesarios para modernizar los servicios que ésta presta a los estudiantes. Las principales características de “Requenguin” se pueden definir de la siguiente manera:

- El juego proporciona un entorno que permite a los estudiantes identificar las clases de stakeholders que tienen un interés legítimo en el software que se pretende desarrollar.
- El juego provee a los estudiantes de capacitación en técnicas de entrevista basadas en un metamodelo lingüístico para obtener requisitos de los stakeholders.
- El juego también proporciona capacitación a los estudiantes para definir los requisitos del software.

- El juego proporciona apoyo y orientación a los estudiantes para definir restricciones, identificar servicios, establecer la interacción entre los usuarios y el software, e identificar otros requisitos relacionados con las cualidades críticas.
- El juego ayuda a los estudiantes a comprender cómo analizar y mantener los requisitos de los stakeholders.



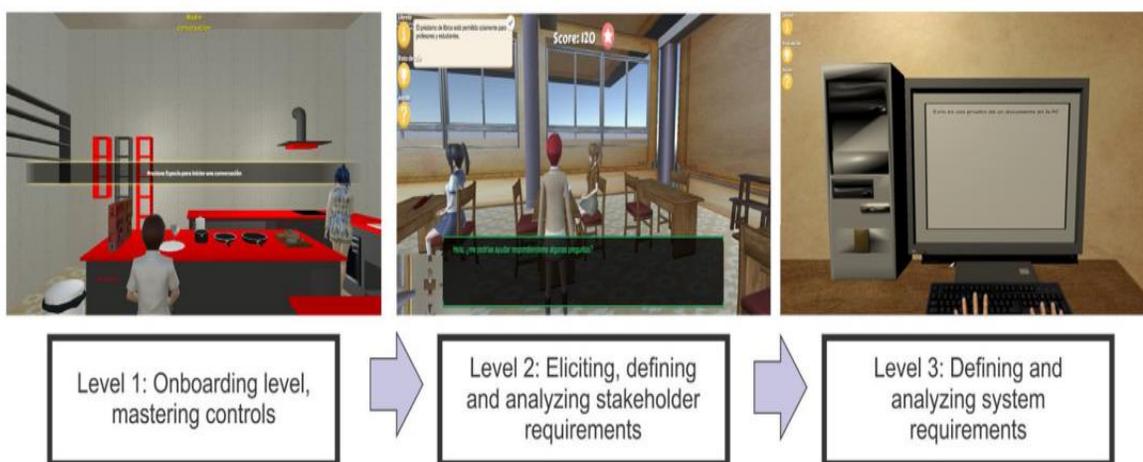
**Figura 2.6.** Pantalla inicial de “Requengin” (tomada de García et al., 2020)

“Requengin” solamente tiene un grado de complejidad que brinda apoyo a los estudiantes a través de sugerencias, consejos, alertas y comentarios durante la sesión de juego. Los estudiantes son atraídos mediante un sistema de recompensas que aumentan su motivación, retención de conocimientos, y aplicación del aprendizaje. Los estudiantes pueden seguir su progreso en las actividades del juego utilizando la información de recompensas proporcionada por el juego, lo que les permite determinar su posición comprada con otros jugadores. El juego incorpora a 53 NPC con los que los estudiantes deben interactuar activamente para realizar las actividades. 30 de estos personajes son estudiantes, cinco son profesores, cinco son parte del personal de la biblioteca, tres son proveedores de libros, y 10 son personajes que no brindan ninguna información útil a los estudiantes (e.g., personal de seguridad, personal de limpieza y visitantes), que de hecho puede causar confusión con sus respuestas ambiguas. “Requengin” utiliza la perspectiva tradicional de un juego de exploración de mundo abierto, donde los estudiantes deben explorar tres escenarios principales para aprender a realizar los procesos definidos por el estándar: la casa del jugador, los pasillos de la universidad, y una biblioteca académica. La Figura 2.7 muestra que estos escenarios se relacionan con tres niveles de ejecución que se describen de la siguiente manera:

- Primer nivel: El escenario de la casa del jugador fue diseñado para familiarizar rápidamente a los estudiantes con las reglas básicas del juego y establecer las expectativas sobre la dinámica y la historia del mismo. En este nivel los estudiantes aprenden a utilizar los controles del juego, establecer conversaciones con NPC durante la sesión de juego, recopilar información a través de entrevistas, y subir su trabajo para que el profesor lo revise en el entorno simulado.
- Segundo nivel: El escenario de los pasillos de la universidad involucra al estudiante en la realización inicial de las tres actividades definidas por el proceso de definición de los requisitos

de los *stakeholders*: obtener los requisitos de los *stakeholders*, definir los requisitos de los *stakeholders*, y analizar y mantener los requisitos de los *stakeholders*. En este nivel los estudiantes realizan entrevistas a NPC que representan a otros estudiantes y profesores siguiendo la aplicación de un metamodelo lingüístico con el objetivo de definir y documentar un conjunto de requisitos para establecer un conjunto en una línea de base para su análisis posterior.

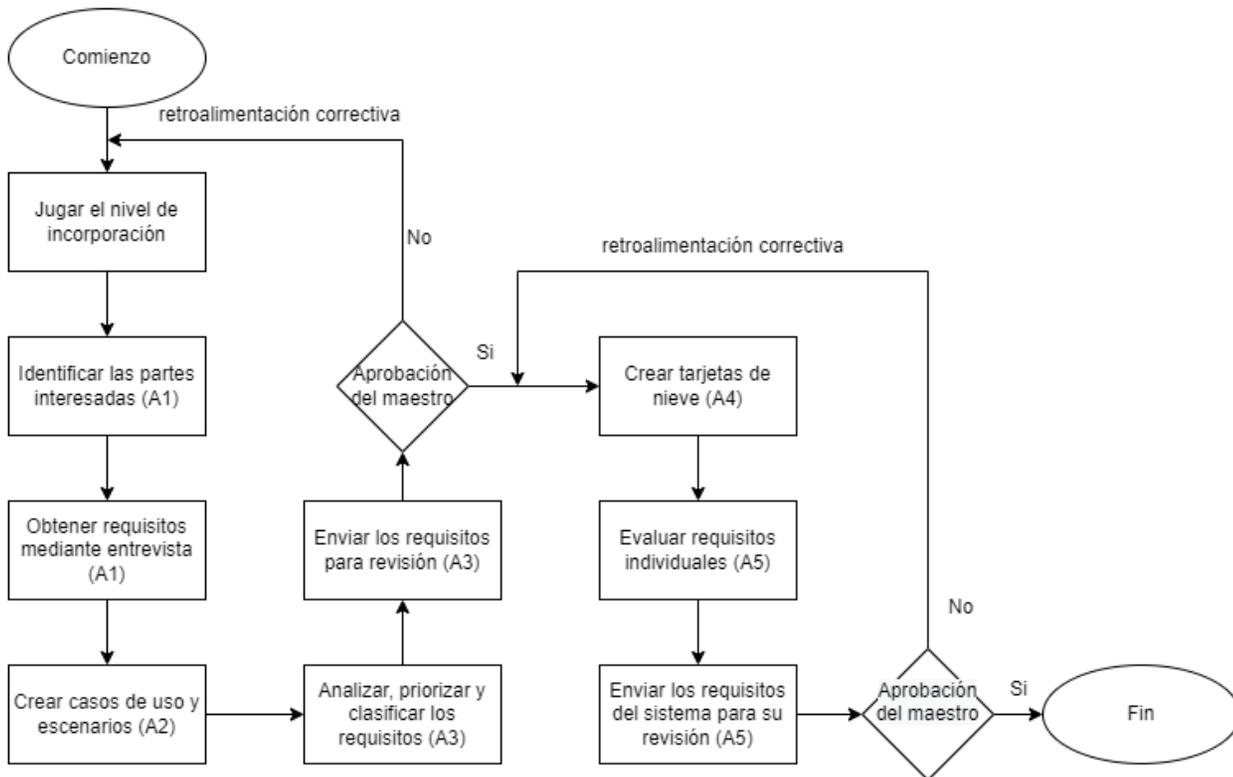
- Tercer nivel: El escenario de la biblioteca involucra a los estudiantes en el desarrollo de las dos actividades del proceso de análisis de requisitos: definir los requisitos del sistema, y analizar y mantener los requisitos del sistema. Aquí se definen los requisitos del sistema y se analizan las características individuales de calidad de los requisitos utilizando las técnicas diagramación de casos de uso, obtención de escenarios, y definición de *snow cards*.



**Figura 2.7.** La estructura de los niveles de “Requengin” (tomada de García et al., 2020)

La Figura 2.8 muestra los pasos definidos en el juego para que los estudiantes recorran los escenarios y logren los objetivos de aprendizaje establecidos para cada uno. Primero, es necesario completar el nivel de incorporación para aprender, mediante una experiencia de aprendizaje práctico, cómo usar los controles para realizar una acción en el juego. Es obvio que el nivel de incorporación se trata de enseñar a los estudiantes las reglas y herramientas del juego. Por lo tanto, el nivel de incorporación comienza tan pronto como los estudiantes se registran y finaliza cuando dominan las habilidades fundamentales necesarias para desenvolverse correctamente en el juego y lograr los puntos requeridos para acceder a la biblioteca. Además, la estructura de recompensas definida para el juego se habilita desde la etapa de nivel de incorporación y promueve la motivación de los estudiantes en los otros niveles. También se implementó la técnica de programación neurolingüística para capacitar a los estudiantes en la realización de una entrevista productiva utilizando patrones lingüísticos. En este sentido, el juego tiene un conjunto de preguntas y respuestas precargadas que los estudiantes deben seleccionar. Por lo tanto, los estudiantes son notificados durante la entrevista cuando se identifica un patrón y pueden desencadenar una nueva pregunta que puede producir una nueva respuesta. Las nuevas preguntas también aumentan la puntuación de los estudiantes. El juego incorpora cuatro personajes jugables más para proporcionar a los jugadores nueva información sobre el dominio del problema y alentar a los estudiantes a inventar nuevos requisitos durante la actividad de elicitación. Tres compañeros reciben una explicación previa sobre los deseos y necesidades del sistema de la biblioteca para participar en las sesiones de juego de tal manera que (1) brindan información que permite a los jugadores inventar nuevos requisitos o (2) brindan información confusa que los estudiantes deben considerar en el análisis posterior para identificar intereses legítimos. Se le

da un personaje más a un profesor para que interactúe con los estudiantes durante la sesión de juego según sea necesario.



**Figura 2.8.** Flujo de juego de “Requengin” (traducida de García et al., 2020)

### 2.4.4.3. Resultados

Se realizó una evaluación empírica con 20 estudiantes del tercer año de la Licenciatura en Informática, quienes tomaban el curso de Ingeniería de Software en NovaUniversitas. Estos estudiantes respondieron dos cuestionarios después de llevar a cabo las sesiones de juego con el fin de proporcionar información sobre sus percepciones acerca del mismo. El primer cuestionario consideró a MEEGA para evaluar tres categorías mediante 27 ítems: motivación y experiencia del estudiante, y resultados del aprendizaje. Por otro lado, se diseñó un cuestionario adicional de seis ítems para evaluar las experiencias de aprendizaje percibidas por los estudiantes tras finalizar su interacción con “Requengin”. En el estudio también participaron tres profesores con más de 12 años de experiencia en la enseñanza de la Ingeniería de Software a nivel licenciatura, quienes observaron a los estudiantes, les brindaron realimentación y orientación, y analizaron su desempeño durante las sesiones de juego. Por lo tanto, se diseñó un cuestionario final de ocho ítems para recopilar también información de estos tres profesores. La evaluación inició con la impartición de clases teóricas para proporcionar a los estudiantes los conceptos fundamentales del estándar ISO/IEC/IEEE 29148:2011. Estas clases se centraron en proveerles una referencia detallada sobre los procesos de Ingeniería de Requisitos definidos por el estándar y el uso adecuado de algunas técnicas de Ingeniería de Requisitos que facilitan su implementación. Posteriormente, se realizaron sesiones de juego con “Requengin” para que los estudiantes aplicaran los conceptos aprendidos en el curso teórico. Finalmente, los datos se recopilaron y luego se analizaron utilizando estadísticas descriptivas para identificar los aspectos

positivos y negativos del juego. Las principales conclusiones de la evaluación se resumen a continuación.

En términos generales, los estudiantes se mostraron bastante motivados al jugar con “Requenguin” con el objetivo de fortalecer los conceptos aprendidos durante las clases teóricas. En este sentido, los estudiantes manifestaron que era bastante común que no tomaran en serio a la Ingeniería de Requisitos, puesto que el uso excesivo de las lecciones teóricas sobre este tópico a menudo les generaba aburrimiento. Los estudiantes también argumentaron que, como consecuencia de lo anterior, regularmente se le presta mayor atención a la programación de sistemas. En resumen, la información recopilada permitió hallar evidencia de que “Requenguin” podría contribuir a mejorar la forma en que la Ingeniería de Requisitos es enseñada a nivel de licenciatura.

En cuanto a las percepciones de los estudiantes, éstos consideraron que las sesiones de juego fueron adecuadas y estimulantes, y que les ayudaron a fortalecer su comprensión de los conceptos teóricos aprendidos en el aula. Casi todos los ítems en la categoría de experiencia lograron una media apropiada, con la excepción de “*III. Olvidé mis preocupaciones diarias y me concentré mientras jugaba con Requengin*” e “*I20. Cuando interrumpieron la sesión de juego, me decepcionó que mi tiempo de jugar con Requengin hubiera terminado*”, que registraron los valores más bajos. Estos datos evidenciaron que las dinámicas, mecánicas, y apariencia de “Requenguin” debían mejorarse para obtener mejores resultados. Por otro lado, los estudiantes consideraron que dicho juego les proporcionó niveles adecuados de desafío y competencia al jugar para mejorar su comprensión de los fundamentos definidos en el estándar.

Sin duda la categoría mejor calificada fue la de resultados de aprendizaje, donde los tres *ítems* evaluados alcanzaron el valor máximo de 2. Todos los estudiantes sintieron que “Requenguin” podía ser una herramienta de enseñanza/aprendizaje eficiente para fortalecer sus conocimientos sobre los conceptos básicos establecidos por el estándar ISO/IEC/IEEE 29148:2011. Además, los estudiantes sintieron que el juego les permitió comprender de manera eficiente el estándar, en comparación con otros enfoques tradicionales. Más importante aún, los estudiantes sintieron que la experiencia derivada de aprender con el juego les ayudará en su vida profesional. Como consecuencia, las dimensiones aprendizaje a corto plazo y aprendizaje a largo plazo recibieron puntuaciones excelentes.

Con relación a los ítems adicionales diseñados para evaluar la experiencia de aprendizaje percibida por los estudiantes, el 91.25% afirmó no haber tenido dudas de que “Requenguin” contribuyó positivamente en su formación al mejorar su comprensión de los conceptos relacionados con la identificación correcta y productiva de los *stakeholders* para obtener los requisitos adecuados del sistema, definición de los requisitos de los *stakeholders* y los requisitos del sistema, análisis de los requisitos de los *stakeholders* y de los requisitos del sistema, así como la definición de requisitos funcionales y no funcionales. En resumen, los estudiantes opinaron que “Requenguin” contribuyó a mejorar la comprensión de los procesos y actividades definidos por el estándar ISO/IEC/IEEE 29148:2011.

Finalmente, la evaluación sobre la aceptación y expectativas de los profesores sobre “Requenguin” fue positiva. Durante la evaluación se recogió información de que el 100% de éstos manifestó su interés en sugerir el uso de “Requenguin” como herramienta didáctica complementaria a sus colegas que no participaron en el estudio. De igual forma, todos los profesores también coincidieron en que “Requenguin” era fácil de entender y jugar y que podría considerarse una buena alternativa para apoyar el enfoque educativo en la enseñanza de los fundamentos del estándar ISO/IEC/IEEE 29148:2011 y los procesos de la Ingeniería de Requisitos a nivel licenciatura.

## 2.4.5. Estrategia basada en un juego serio para la capacitación a nivel licenciatura en el estándar ISO/IEC 29110:2018

### 2.4.5.1. Objetivo

El juego serio “Tablero ISO/IEC 29110” (Bonilla-Rivas et al., 2021) fue desarrollado como una estrategia alternativa de enseñanza entre usuarios que poseen nulo o un mínimo conocimiento sobre la estructura y elementos que conforman el perfil básico del estándar ISO/IEC 29110:2018 *Systems and software engineering – Lifecycle profiles for very small entities*. Por lo tanto, dicho juego pretendió mejorar el entrenamiento sobre el estándar en un escenario donde existe la carencia de instrumentos de enseñanza que mejoren el conocimiento de estudiantes a nivel licenciatura, quienes posiblemente deban realizar la implementación del estándar en sus vidas profesionales.

### 2.4.5.2. Descripción

Como se mencionó anteriormente, el juego serio proporciona la información de las actividades, los productos de trabajo (i.e., entrada, internos o de salida), así como de los roles que deben intervenir en los dos procesos definidos por el estándar ISO/IEC 29110:2018: “Gestión del Proyecto” e “Implementación de Software”. Por lo tanto, dicho juego considera dos tableros, uno por cada proceso definido por el estándar, para implementar la mecánica del juego de la Oca con el objetivo de que los jugadores avancen por las casillas hasta llegar a la meta. Su desarrollo siguió una estrategia basada en ocho puntos que se describen a continuación:

1. Definir un objetivo, el cual se centró en la obtención de un aprendizaje específico.
2. Definir a quien se pretendía dirigir el juego con el objetivo de garantizar una experiencia fluida y envolvente. En este sentido, el diseño del juego se basó en responder las siguientes preguntas: ¿sobre qué será?, ¿para quién se desarrollará?, ¿dónde se va a utilizar?, ¿cuándo se jugará? y ¿cómo se va a jugar?
3. Promover la incorporación y el compromiso de los jugadores mediante tareas que disfruten realizar. Para cumplir con este punto se utilizaron elementos de gamificación, tales como tablas de clasificación, sistemas de puntos, insignias, niveles, barras de progreso, recompensas, puntuaciones, desafíos, logros, realimentación y desbloqueo de contenido.
4. Crear un documento con el diseño preliminar que sirva como referencia en la producción del juego serio, el cual deberá estar bien descrito con lujo de detalles. En este sentido, el Documento de Diseño del Juego (GDD, por sus siglas en inglés) fue utilizado para describir el cuerpo y el alma del juego, definir las prioridades y analizar las alternativas posibles de implementación. GDD implementa la especificación de requisitos de software mediante la aplicación de un marco de diseño de juegos estándar llamado MDA (Mecánica, Dinámica, Estética).
5. Realizar un ciclo de diseño en el cual se defina un primer concepto y se obtenga un primer prototipo. Dicho prototipo fue probado con usuarios seleccionados para obtener realimentación que permitió perfeccionar el prototipo y convertirlo en una mejor propuesta. El proceso se repitió en repetidas ocasiones hasta concluir la corrección de errores y reducir el número de mejoras.
6. Desarrollar del juego serio a partir de una idea general. En este sentido, se diseñó la jugabilidad, se creó un prototipo de software y se probó con el fin de generar nuevas versiones.
7. Verificar la percepción del usuario y la experiencia obtenida al jugar con el objetivo de implementar mejoras al juego serio.

8. Evaluar la información recolectada sobre el aprendizaje percibido, la usabilidad del juego, la confidencialidad, la satisfacción, la diversión, la atención y la percepción del juego que el usuario experimentó durante la sesión de juego.

Considerando lo anterior, se diseñó un juego digital que introduce dos tableros para presentar una visión general de los procesos de “Gestión del Proyecto”, el cual contiene un total de 28 casillas, e “Implementación de Software”, que contiene un total de 37 casillas. Ambos tableros cuentan con casillas de comodines que ayudan a los jugadores a avanzar de manera más rápida, y casillas de retos que muestran preguntas aleatorias de acuerdo con el recorrido del jugador (véase Figura 2.9).



**Figura 2.9.** Pantalla del tablero de “Gestión del Proyecto” (tomada de Bonilla-Rivas et al., 2021)

El juego se desarrolló utilizando las siguientes herramientas:

- *Genially*: Permitió la configuración del juego en un tiempo corto mediante el uso de imágenes interactivas, guías, experiencias gamificadas y otros recursos.
- *Dreamweaver*: Permitió la creación de páginas web para visualizar las preguntas que se muestran en el juego.
- *Amazon Web Services (AWS)*: Permitió el alojamiento tanto de las páginas que contienen las preguntas a responder durante el juego como del audio descriptivo de cada casilla.

Básicamente el juego introdujo la siguiente dinámica. Los jugadores colocan una ficha que los representa en la casilla inicial y se turnan para lanzar el dado. De acuerdo con el turno de cada jugador, se lanzará el dado y el número que resultante será el número de casillas que el jugador deberá avanzar en el tablero. Un jugador podrá visualizar una descripción de la casilla si da clic sobre ella. Además, podrá escuchar un audio con la descripción. Si algún jugador necesita leer las instrucciones del juego, éste lo podrá hacer dando clic en la imagen de instrucciones en el tablero del juego serio. Por otro lado, al lanzar los dados puede darse cualquiera de estas situaciones:

- Si un jugador cae en una casilla de color rojo, éste deberá dar clic en la casilla para responder una pregunta. En caso de que el jugador responda correctamente, permanecerá en dicha casilla, de lo contrario tendrá que regresar a la casilla que indique la pregunta y ver su descripción.

- Si un jugador cae en una casilla de color verde oscuro todos los jugadores deberán avanzar una casilla extra.
- Si un jugador obtiene un 5 al tirar el dado, éste avanzará una casilla extra.
- El jugador que llegue primero a la casilla final es el ganador del juego.

Cuando los jugadores logren terminar el tablero de “Gestión del Proyecto”, se debe dar clic en el botón “jugar” del proceso de “Implementación de Software” para visualizar el tablero correspondiente.

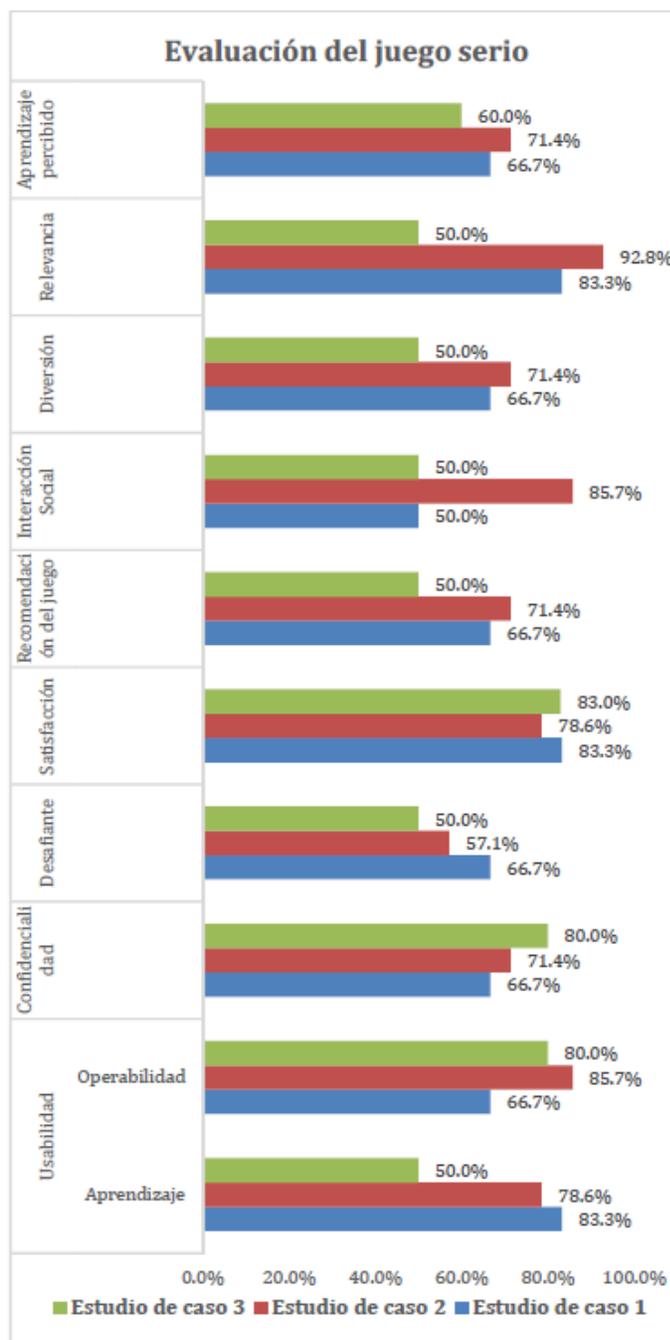
### 2.4.5.3. Resultados

La evaluación empírica del juego se realizó mediante tres casos de estudio que contaron con la participación de estudiantes de nivel licenciatura de la Universidad de Guadalajara, México, y de la Universidad de Medellín, Colombia. En este sentido, se utilizó *Google Forms* para aplicar dos cuestionarios que permitieron la obtención de estadísticas sobre la opinión de los estudiantes. El primer cuestionario recopiló información sobre el nivel de conocimiento que los jugadores tenían sobre el estándar, mientras que el segundo recopiló información sobre el aprendizaje logrado por los jugadores y, al mismo tiempo, la evaluación del juego serio. Además, se incluyó una sección enfocada en la evaluación del juego serio siguiendo el modelo de evaluación MEEGA+, con el objetivo de analizar el juego educativo y evaluar la calidad percibida por los estudiantes en un contexto educativo. Los resultados se muestran en la Figura 2.10.

Al aplicar el primer cuestionario, que se enfocó a determinar el nivel de conocimiento de los estudiantes sobre el estándar, se obtuvo que para los participantes del caso de estudio 1 el 83% de los mismos tenía un mínimo conocimiento del estándar, el 50% tenía una idea sobre el tipo de empresas al que se dirige, y el 83% no conocía su estructura. Después de que estos estudiantes utilizaran el juego serio se obtuvieron resultados que evidenciaron una mejora. Es decir, las respuestas al segundo cuestionario mostraron que el 100% de los alumnos fue capaz de identificar el tipo de empresas al que se dirige el estándar, el 50% identificó claramente los elementos que componen al estándar (e.g., procesos, productos de trabajo, y roles) y el 83% identificó de manera correcta la estructura del estándar.

De la misma manera, los participantes del caso de estudio 2 mostraron una mejora similar puesto que como resultado de la aplicación del primer cuestionario se obtuvo que el 56% de los estudiantes no tenía conocimiento del estándar, el 33% tenían una idea sobre el tipo de empresas al que se dirige el estándar, y el 89% no sabían cómo estaba estructurado. Después de que los alumnos realizaran sesiones de juego se aplicó el segundo cuestionario para obtener resultados que mostraron una mejora importante puesto que el 77% de los estudiantes identificó correctamente el tipo de empresas a las que se dirige el estándar, el 57% identificó los elementos que componen al estándar, mientras que el 71% identificó también de manera correcta su estructura.

Finalmente, los participantes del caso de estudio 3 también mostraron una mejora significativa puesto que inicialmente solamente el 50% de los mismos identificaban el tipo de empresas al que se dirige el estándar y conocían su estructura. No obstante, después de participar en las sesiones de juego, el 67% de los estudiantes fue capaz de identificar el tipo de empresas a las que está dirigido el estándar, el 50% identificó correctamente los elementos que lo componen, y el 100% identificó de manera correcta su estructura, confirmándose así que los conocimientos sobre el estándar ISO/IEC 29110:2018 fueron reforzados/adquiridos mediante el uso del juego serio.



**Figura 2.10.** Resultados de la evaluación empírica (tomada de Bonilla-Rivas et al., 2021)

Con estos resultados, la investigación concluye que los juegos serios se han posicionado como una alternativa para motivar la enseñanza en diversos dominios, incluyendo la Ingeniería de Software, permitiendo transformar el enfoque tradicional de enseñanza en un enfoque más atractivo, tanto para formar a los estudiantes como para la formación en la industria del desarrollo de software. Estos resultados evidenciaron que el uso del juego serio “Tablero ISO/IEC 29110” contribuyó a mejorar la enseñanza y/o reforzó el conocimiento sobre el estándar internacional ISO/IEC 29110:2018. Sin embargo, también se han identificado oportunidades de mejora como son: añadir una selección de

dificultad, establecer un límite de tiempo para responder las preguntas y permitir que varios usuarios jueguen a distancia sin necesidad de compartir la misma pantalla.

## 2.5. Consideraciones finales sobre el estado del arte

Como se puede observar en los trabajos relacionados, la enseñanza de los estándares del proceso de software ha representado una tarea desafiante. Los métodos tradicionales de enseñanza no han dado resultados positivos debido a su dependencia de cursos altamente teóricos. Esta dependencia genera aburrimiento en los estudiantes y termina propiciando bajos niveles de motivación por aprender la estructura de un estándar, su correcta aplicación, y su importancia en la industria moderna del software. La principal consecuencia de esta situación es que los universitarios egresados se encuentran con complicaciones al momento de encontrar un trabajo con buen salario, puesto que muchas de las veces no poseen las habilidades duras y blandas mínimas requeridas por la industria.

Sin embargo, aunque es evidente que los métodos tradicionales de enseñanza no han generado resultados positivos en la enseñanza de tópicos particulares de la Ingeniería de Software, las universidades tampoco han prestado la atención necesaria en desarrollar e implementar métodos alternativos de enseñanza que propicien mejores resultados, como es el caso de los juegos serios. Como se observó en este capítulo, los juegos serios introducen la posibilidad de crear entornos de aprendizaje más atractivos, inmersivos e interactivos para los estudiantes, puesto que generan un alto porcentaje de aceptación de la comunidad de especialistas tanto de la academia como de la industria. Estos entornos simulados permiten al estudiante interactuar y relacionarse de forma más realista con los distintos procesos y estándares que son utilizados en el desarrollo de software, consolidando así su conocimiento y dándole la confianza de poder equivocarse sin generar un error catastrófico. Además, debido a la diversidad de tipos de juegos que se pueden desarrollar (e.g., juegos de tablero, juegos de puzzles, juegos de acertijos, entre muchos otros), se puede lograr una implementación personalizada para prácticamente cada proceso y estándar que se utilice en el mundo real. Por lo tanto, al adoptar una tecnología que es altamente atrayente para los estudiantes universitarios se consigue una mayor motivación y una mejoría en la atención y retención de los conocimientos.

De hecho, en la actualidad se puede encontrar una cantidad importante de propuestas de juegos serios enfocados a la enseñanza de los conceptos elementales de distintos estándares del proceso de software, los cuales ya han sido evaluados empíricamente para confirmar la obtención de resultados positivos en estudiantes recién egresados que se integran a la industria. Sin embargo, aún quedan muchos estándares y procesos de importancia en la actualidad que no han sido abordados por un juego serio, por lo que es crucial el explorar diferentes alternativas de enseñanza para que los estudiantes adquieran el conocimiento elemental sobre éstos. Por lo tanto, es recomendable que las universidades y los educadores integren herramientas tecnológicas modernas que sirvan de apoyo para reforzar y complementar el conocimiento adquirido en las aulas de clases y que permitan, al mismo tiempo, que los estudiantes experimenten un acercamiento más realista a los distintos procesos y estándares del proceso de software que son utilizados actualmente por la industria real. De esta manera se estaría contribuyendo a reducir la brecha de conocimiento que existe entre el conocimiento que un universitario recién egresado posee y el que la industria realmente demanda.

En este sentido, en el siguiente capítulo de esta tesis presenta la propuesta de un juego serio para enseñar los elementos fundamentales del estándar ISO/IEC/IEEE 26515:2018 - *Systems and software engineering —Developing information for users in an agile environment*.

### 3. Diseño y desarrollo del juego serio

En este capítulo se presentan las decisiones metodológicas que fueron tomadas para diseñar e implementar el juego serio propuesto en esta tesis para mejorar la educación sobre el estándar ISO/IEC/IEEE 26515:2018 a nivel licenciatura. Aunado a esto, se presentan las principales funcionalidades del juego con el fin de demostrar la cobertura de los elementos fundamentales de dicho estándar.

#### 3.1. Aspectos metodológicos

El desarrollo y la implementación del juego propuesto en esta tesis se llevó a cabo siguiendo los principios de la Metodología para el Desarrollo de Juegos Educativos basados en Guiones Interactivos de Lope et al., (2015). Dicha metodología propone un conjunto de fases y actividades destinadas a la elaboración de un juego educativo (véase Figura 3.1).

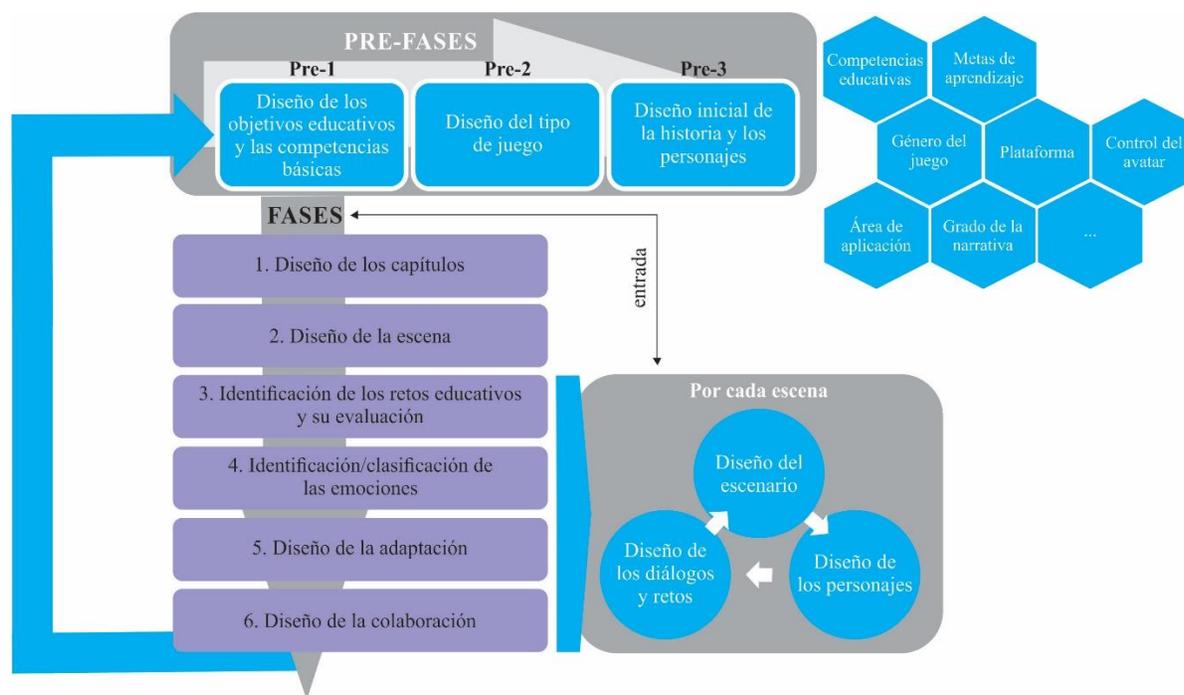


Figura 3.1. Metodología para desarrollar el juego serio (de Lope et al., (2015), como se citó en (León, 2016))

En este sentido, la metodología establece que inicialmente se deben completar tres pre-fases que abarcan los siguientes aspectos: la *conceptualización de los desafíos educativos*, la *definición y diseño del tipo de juego*, y la *creación de una narrativa inicial y personajes principales*. Posteriormente, se procede a realizar una etapa compuesta por seis fases, las cuales marcan el inicio del proceso de desarrollo del juego. Este ciclo se repetirá de manera iterativa hasta que se determine que el juego cumple con las expectativas establecidas. Además, este enfoque metodológico plantea la elaboración de un guion narrativo del juego, el cual se debe estructurar en capítulos y escenas. Después, de forma gradual, se incorporan los componentes restantes del juego (i.e., entornos, personajes, entretenimiento, desafíos). A continuación, se describen las distintas actividades propuestas por esta metodología para realizar las pre-fases mencionadas:

- *Pre-1: Diseño de los objetivos educativos y las competencias básicas.* En esta primera fase preliminar el equipo de educadores y profesores deberán determinar las competencias y los objetivos educativos específicos que serán abordados por el juego.
- *Pre-2: Diseño del tipo de juego.* Para realizar la segunda pre-fase es necesario determinar las características del juego que puedan influir en las decisiones futuras de diseño. Entre estas características se incluyen el género del juego, el control de avatar, la plataforma para la que se desarrollará, quienes serán los futuros usuarios, el nivel narrativo, y el área de aplicación e interactividad.
- *Pre-3: Diseño inicial de la historia y los personajes.* Para desarrollar la historia del juego se requiere de la realización de varias iteraciones proporcionales al nivel narrativo, por lo que una narrativa muy compleja puede complicar el mantener la visión general de la aventura y las dinámicas que se hayan elegido para el juego. Para evitar cualquier complicación futura se sugiere elaborar un diseño inicial de la historia (pudiendo apoyarse con bocetos gráficos) y definir algunos personajes principales del juego a medida que ésta se desarrolla.

Una vez que se hayan finalizado las fases preliminares, se procede a realizar las siguientes fases:

- **Fase 1: Diseño de los capítulos.** En el contexto de la metodología, un capítulo se define como el ítem de más alto nivel utilizado para organizar la historia, de tal forma que se facilite la integración del contenido. En este sentido, es obligatorio que un juego contenga al menos un capítulo, aunque es más común que se componga de varios de ellos. Cada uno de los capítulos se define por un identificador, un nombre abreviado, y la trama general de la aventura que se desarrollará.
- **Fase 2: Diseño de la escena.** Cada capítulo definido en la primera fase será dividido en escenas que conformarán su trama y que serán descritas por un nombre único y un breve resumen. Es necesario definir el número de escenas y su orden tomando como referencia las decisiones y las acciones que un jugador pueda tomar. Esta libertad en la toma de decisiones producirá escenas opcionales que podrán, o no, ser experimentadas por un jugador. Cada escena especificará los siguientes elementos:
  - Diseño del escenario. Un escenario es el lugar en donde acontecen las acciones y los diálogos de una escena; por lo tanto, su definición incluye una parte estática (i.e., el entorno y los objetos que se encuentran en él) y otra parte dinámica (i.e., la interactividad de los objetos).
  - Diseño de los personajes. Cada escena incluirá uno o más personajes en el escenario, por lo que es necesario describir a detalle sus caracteres, apariencias, y personalidades desde la primera vez que aparezcan. Aunque en un principio dicha descripción se

realiza con palabras, en iteraciones futuras se puede hacer uso de bocetos gráficos como ayuda.

- Diseño de los diálogos y desafíos del juego. Cuando una escena esté transcurriendo, los personajes deberán realizar acciones específicas para superar los desafíos que el juego presente, apoyándose de la comunicación y el diálogo. Para describir estos elementos de la escena se pueden utilizar diagramas de flujo. Con el objetivo de aumentar la diversión, la complejidad, y la inmersión del jugador es necesario que el diálogo no se cierre, si no que dependa directamente de las respuestas que el jugador proporcione. Estas variantes de decisiones se deberán especificar también en el diagrama de flujo del diálogo.
- **Fase 3: Identificación de los retos educativos y su evaluación.** Los objetivos educativos por los cuales se desarrolla un juego serio a menudo se esconden en partes específicas de los diálogos (e.g., como un planteamiento de un desafío educativo o como información necesaria para superar un desafío), además dichos objetivos se asocian con los retos lúdicos del mismo. Para determinar si un objetivo educativo ha sido satisfecho mediante un desafío es necesario definir la regla de evaluación que corresponda, la cual puede incluir condiciones para su aplicación y hacer uso de valores que son recogidos en diferentes partes del escenario o incluso en otros escenarios en donde el jugador haya trabajado en la tarea educativa. En este mismo contexto, es importante señalar que el factor educativo puede ser dividido no solo entre las acciones lúdicas y los diálogos, también se podrá hacer uso de los objetos del escenario con los que un jugador puede interactuar.
- **Fase 4: Identificación/clasificación de las emociones.** Un aspecto que debe ser considerado al diseñar un escenario es la identificación de las diversas emociones que se pretende generar en un jugador. Estas emociones deben ser clasificadas marcando las partes del diálogo o la serie de pasos de una acción en la cual se desea generar una reacción por parte del jugador, permitiendo así el diseño de la experiencia del jugador.
- **Fase 5: Diseño de la adaptación.** Durante esta fase es obligatorio determinar si el juego tiene la capacidad de adaptarse a las capacidades y características del jugador, del dispositivo de juego, o del entorno de juego. Debido a esto, es necesario definir los atributos que se pueden personalizar en el juego (e.g., desafíos educativos, modo de interacción, narrativa, criterios de evaluación) en función de las propiedades (e.g., resolución del dispositivo de juego, contexto físico, conocimiento previo, preferencias del jugador). Además de esto, se debe especificar cómo se deben realizar los ajustes (e.g., diversas técnicas de adaptación que influyen en la dificultad y el cambio de apariencia de un personaje) y cuándo y cómo el jugador controlará dichos ajustes.
- **Fase 6: Diseño de la colaboración.** En la última fase es obligatorio indicar qué acciones o serie de pasos, si las hubiera, deberán realizarse en grupo.

De esta forma, el diseño del juego planteado en esta tesis siguió el proceso mencionado previamente. En consecuencia, las próximas secciones de este capítulo ofrecerán una exposición detallada sobre la aplicación de la metodología mencionada anteriormente.

### 3.1.1. Pre-1: Diseño inicial de los objetivos educativos y las competencias básicas

Dado que el estándar ISO/IEC/IEEE 26515:2018 se enfoca principalmente en la definición de tareas y actividades para el desarrollo de documentación para usuarios en entornos ágiles, la realización de esta fase propuso el uso de entrevistas y encuestas a profesores que imparten clases

sobre Ingeniería de Software relacionadas con el uso de los métodos ágiles. Por consiguiente, y dado que no se pretendió crear un nuevo curso para la UTM, posteriormente se llevó a cabo una revisión detallada de los planes de estudio actuales con los que cuentan las distintas universidades del Sistema de Universidades Estatales de Oaxaca (SUNEO) y su contenido respecto a los métodos ágiles para identificar los retos y objetivos educativos que el juego debía abordar. Es decir, los planes de estudios de la Licenciatura en Informática e Ingeniería en Computación de la UTM, la Universidad del Istmo (UNISTMO), la Universidad del Papaloapan (UNPA), la Universidad del Mar (UMAR), la Universidad de la Sierra Sur (UNSI), la Universidad de la Sierra Juárez (UNSIJ), la Universidad de la Cañada (UNCA), y NovaUniversitas fueron incluidos en este análisis.

### 3.1.1.1. Recolección de opiniones y profesores sobre los métodos de enseñanza

Por lo tanto, en primera instancia se diseñaron y aplicaron dos cuestionarios dirigidos a profesores y estudiantes de dichas universidades. Es importante mencionar que la participación fue voluntaria y se logró contar con la colaboración de 35 profesores y 80 estudiantes (inscritos en octavo y décimo semestres) del SUNEO. Por razones de confidencialidad los datos personales de los participantes no fueron recogidos durante la aplicación de los cuestionarios. Además, es necesario enfatizar que tales instrumentos de evaluación son adaptaciones de los propuestos por León (2016) para determinar la pertinencia de introducir el Aprendizaje basado en Juegos (GBL, por sus siglas en inglés) en la enseñanza de la Ingeniería de Software a nivel licenciatura, específicamente para la enseñanza de la elicitación de requisitos de software. Tales cuestionarios se basan en una escala de evaluación tipo Likert (1932), la cual consta de cinco niveles de respuesta. Esta elección se basó en la característica distintiva de esta escala, la cual radica en la formulación de una serie de enunciados (i.e., ítems) cuidadosamente seleccionados según el ámbito de análisis. Estos enunciados abarcan una gama de opciones que van desde “Totalmente de acuerdo” hasta “Totalmente en desacuerdo”. Así pues, los participantes de la encuesta deben evaluar los ítems planteados siguiendo un patrón uniforme de respuesta que facilita la rápida comprensión de la mecánica de respuesta.

**Tabla 3.** Cuestionario para opiniones de los profesores sobre los métodos de enseñanza (adaptada de León, 2016)

	Ítem	Escala de respuestas					Calificación	
		TA	DA	MI	ED	TD	M	DS
Metodologías tradicionales	P1. Las metodologías tradicionales de enseñanza que se centran en el uso exclusivo de lecciones teóricas muestran una limitada efectividad al aplicarse en campos altamente prácticos como el uso de los métodos ágiles.	27	8	0	0	0	1.8	0.4
	P2. Las metodologías tradicionales de enseñanza promueven el desarrollo de habilidades prácticas sobre los métodos ágiles y en los procesos que se utilizan para generar la documentación requerida por los mismos.	0	0	0	31	4	-1.1	0.3
	P3. Las clases predominantemente teóricas generan el estímulo adecuado para despertar el interés de los estudiantes en el aprendizaje de los métodos ágiles y en los procesos necesarios para generar la documentación requerida.	0	0	0	24	11	-1.3	0.5
	P4. Las metodologías tradicionales de enseñanza promueven la interacción y el intercambio de opiniones sobre los temas y, como consecuencia, facilitan la realimentación del conocimiento adquirido.	9	20	0	6	0	0.9	1.0
	P5. Las metodologías tradicionales de enseñanza tienen la capacidad de captar la atención de los estudiantes únicamente durante un lapso breve de tiempo, posiblemente no mayor a los 15 minutos.	0	0	0	3	32	-1.9	0.3

	Ítem	Escala de respuestas					Calificación	
		TA	DA	MI	ED	TD	M	DS
	P6. Mediante el uso de las metodologías tradicionales de enseñanza se logra establecer una conexión auténtica entre los estudiantes y los desafíos que podrían surgir en un entorno laboral donde se utilicen métodos ágiles.	1	8	0	26	0	-0.5	0.9
	P7. Las metodologías tradicionales de enseñanza promueven el desarrollo de las habilidades suaves e interpersonales esenciales requeridas para los profesionales de la industria actual que requieran utilizar métodos ágiles.	0	0	2	28	5	-1.1	0.4
Juegos educativos	P8. Los procedimientos y enfoques para generar la documentación requerida por los métodos ágiles deben ser susceptibles de ser utilizados por los estudiantes en situaciones reales y/o simuladas a través de, por ejemplo, juegos educativos.	1	34	0	0	0	1.0	0.2
	P9. La participación y el dialogo entre los estudiantes dentro de un juego educativo ofrece un estímulo efectivo para su proceso de aprendizaje.	25	9	1	0	0	1.7	0.5
	P10. El uso de juegos educativos puede aumentar la disposición de los estudiantes para aprender, al mismo tiempo que incrementa su motivación.	10	8	0	10	7	0.1	1.6
	P11. Mediante el uso de juegos educativos se puede lograr una aproximación más efectiva de los estudiantes al contexto del mundo real.	28	7	0	0	0	1.8	0.4
	P12. Los juegos educativos pueden contribuir al desarrollo de competencias esenciales en los estudiantes, preparándolos de manera efectiva para su integración en el ámbito laboral.	14	21	0	0	0	1.4	0.5
	P13. El uso de juegos educativos para la enseñanza puede enriquecer la experiencia de aprendizaje de los estudiantes.	0	30	2	2	1	0.7	0.7

En este sentido, el cuestionario dirigido a los profesores está conformado por 13 ítems cuyo objetivo primordial es evaluar sus perspectivas con respecto a las metodologías de enseñanza convencionales, así como la identificación de los desafíos principales que han enfrentado al implementar estas estrategias en el ámbito de la enseñanza de los métodos ágiles y, si fuera el caso, sobre los métodos utilizados en el desarrollo de la información para los usuarios (véase Tabla 3). Simultáneamente, se buscó medir el grado de aceptación de los profesores con relación a los enfoques de enseñanza basados en el GBL, principalmente en la simulación de entornos, y, en consecuencia, recabar sus opiniones y comentarios acerca de este método educativo alternativo. Con este objetivo en mente, se establecieron cinco valores posibles de respuesta, cuyas definiciones son las siguientes: “Totalmente de acuerdo” (TA) – 2, “De acuerdo” (DA) – 1, “Me es indiferente” (MI) – 0, “En desacuerdo” (ED) – -1, y “Totalmente en desacuerdo” (TD) – -2. Con el objetivo de transformar las valoraciones subjetivas en datos cuantitativos, la media (M) y la desviación estándar (DS) fueron calculadas. Es importante mencionar que, antes de aplicar el cuestionario, se les pidió a los 35 profesores participantes que definieran el término “metodología tradicional de enseñanza” considerando el contexto particular de los cursos sobre Ingeniería de Software que han impartido a los estudiantes de la Licenciatura en Informática e/o Ingeniería en Computación en sus respectivas instituciones. De esta manera, se llegó al consenso de que una metodología tradicional es aquella en la que “*un profesor imparte clases teóricas frente a un grupo de estudiantes, las cuales se complementan con ejercicios y/o proyectos prácticos que regularmente son desarrollados y entregados por equipos*”. Desafortunadamente, no hubo un acuerdo sobre la cantidad adecuada de ejercicios y/o proyectos que deberían incluirse en cada curso puesto que la mayoría de los profesores

argumentaron que cada uno es diferente y que, como consecuencia, su complejidad varía considerablemente. En términos generales, las opiniones de los profesores sobre las metodologías tradicionales de enseñanza que se utilizan para abordar el tópico de los métodos ágiles versan sobre dos aspectos: la manera en que éstas se usan y su efectividad para abordar los contenidos propuestos en el plan de estudios. En este sentido, considerando el rango de 2.0 a -2.0 de la escala de respuestas, la mayoría de las opiniones de los profesores se mantuvieron del lado negativo de la escala (véase Figura 3.2). Es decir, el 70.2% de las opiniones evidenció que los profesores no consideran adecuado el utilizar únicamente una metodología tradicional para la enseñanza de los métodos ágiles, mientras que el 29.8% está de acuerdo en utilizar el enfoque convencional.



**Figura 3.2.** Opiniones de los profesores con relación a las metodologías tradicionales de enseñanza

Al analizar detalladamente los datos recogidos con la aplicación del cuestionario, se observaron acuerdos totales para ítems específicos del cuestionario (véase Figura 3.3). Por ejemplo, el 100% de los profesores concordó en que las lecciones puramente teóricas no contribuyen a la enseñanza de los métodos ágiles (*P1. Las metodologías tradicionales de enseñanza que se centran en el uso exclusivo de lecciones teóricas muestran una limitada efectividad al aplicarse en campos altamente prácticos como el uso de los métodos ágiles*,  $M=1.8$ ,  $DS=0.4$ ), mientras que consideran que los juegos serios pueden representar una ayuda importante para esta labor (*P8. Los procedimientos y enfoques para generar la documentación requerida por los métodos ágiles deben ser susceptibles de ser utilizados por los estudiantes en situaciones reales y/o simuladas a través de, por ejemplo, juegos educativos*,  $M=1.0$ ,  $DS=0.2$ ). Además, todos los profesores encuestados argumentaron que los juegos serios son capaces de simular situaciones reales de las cuales se obtengan un aprendizaje más aterrizado a la realidad (*P11. Mediante el uso de juegos educativos se puede lograr una aproximación más efectiva de los estudiantes al contexto del mundo real*,  $M=1.8$ ,  $DS=0.4$ ) y que, incluso, pueden servir de soporte educativo para proporcionar a los estudiantes una educación más integral (*P12. Los juegos educativos pueden contribuir al desarrollo de competencias esenciales en los estudiantes, preparándolos de manera efectiva para su integración en el ámbito laboral*,  $M=1.4$ ,  $DS=0.5$ ).

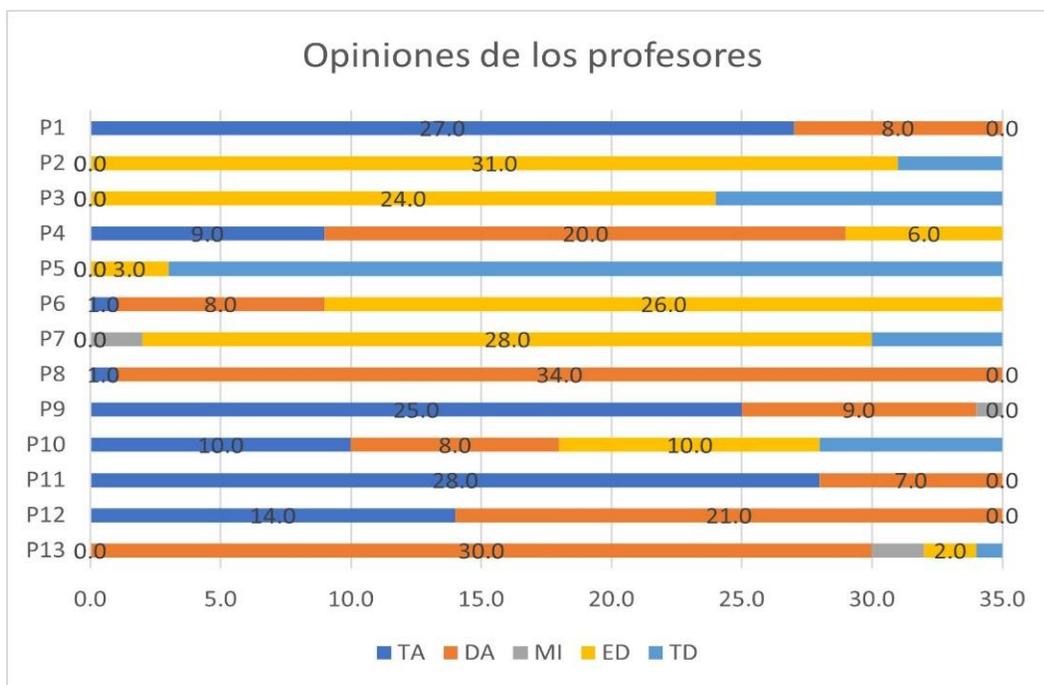


Figura 3.3. Distribución de respuestas en el cuestionario aplicado a los profesores participantes

De manera contraria, el 100% de los profesores argumentó que, considerando un enfoque tradicional de enseñanza, los estudiantes pueden mantenerse atentos a la clase por más de 15 minutos, considerando una duración total de 60 minutos (P5. *Las metodologías tradicionales de enseñanza tienen la capacidad de captar la atención de los estudiantes únicamente durante un lapso breve de tiempo, posiblemente no mayor a los 15 minutos*,  $M=-1.9$ ,  $DS=0.3$ ). En realidad, no se pretendió averiguar por qué esta percepción de los profesores, pero en una plática posterior con éstos se observó una plena confianza en la manera en la que imparten sus cursos. En este sentido, se trata de un aspecto favorable para el desarrollo de esta tesis, puesto que, si este tipo de cursos fuera soportado por una herramienta educativa, se podría mejorar considerablemente el proceso de enseñanza y, por ende, el aprendizaje de los estudiantes. Aunado a lo anterior, la mayor discrepancia entre las respuestas de los profesores se obtuvo cuando se les preguntó si consideraban que un juego serio podría mejorar la disposición y motivación de los estudiantes en sus cursos. Como puede observarse, el valor de  $M=0.1$  para el ítem P10 evidenció que las opiniones de los profesores estuvieron casi equilibradas entre algo positivo y algo negativo, considerando el rango de 2.0 a -2.0. Sin embargo, el valor de  $DS=1.6$  demostró un alto grado de discrepancia en sus respuestas.

Finalmente, el resto de las respuestas de los profesores con respecto a los juegos serios como herramienta de soporte a un curso sobre métodos ágiles fue un tanto diferente, puesto que las medias se mantuvieron del lado positivo de la escala y la discrepancia se mantuvo en un promedio de 0.5. Es decir, los profesores del SUNE0 coincidieron en que este tipo de tecnología podría ayudar a que los estudiantes de nivel licenciatura mejoren su entrenamiento tradicional (véase Figura 3.4).

Por otro lado, un cuestionario de 11 ítems fue utilizado para recoger las opiniones de los estudiantes sobre el uso de las metodologías tradicionales de enseñanza y de los juegos serios como herramientas educativas de soporte para cursos sobre los métodos ágiles (véase Tabla 4). De la misma manera que con el cuestionario de los profesores, se utilizó la escala de cinco valores posibles de respuesta y la media (M) y la desviación estándar (DS) fueron calculadas.



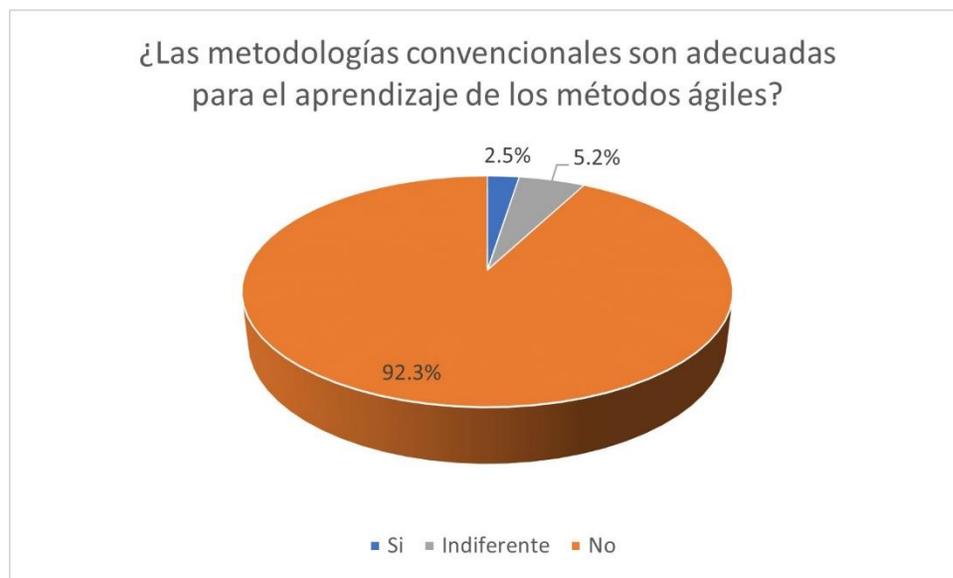
**Figura 3.4.** Opiniones de los profesores con relación a los juegos serios como recurso educativo

**Tabla 4.** Cuestionario para opiniones de los estudiantes sobre los métodos de enseñanza

	Ítem	Escala de respuestas					Calificación	
		TA	DA	MI	ED	TD	M	DS
Metodologías tradicionales	E1. Las metodologías tradicionales de enseñanza que se basan exclusivamente en clases teóricas son monótonas y carentes de interés.	80	0	0	0	0	2.0	0.0
	E2. Las metodologías tradicionales de enseñanza no facilitan la adquisición de experiencia práctica relacionada con la aplicación de conceptos teóricos sobre los métodos ágiles de desarrollo en situaciones reales.	73	7	0	0	0	1.9	0.3
	E3. Las metodologías tradicionales de enseñanza no son adecuadas para fomentar el aprendizaje práctico sobre los métodos ágiles.	11	69	0	0	0	1.1	0.3
	E4. Las clases predominantemente teóricas no generan el entusiasmo necesario para despertar el interés de los estudiantes en el aprendizaje de los métodos ágiles.	40	33	4	3	0	1.4	0.7
	E5. Las metodologías tradicionales de enseñanza no contribuyen en la generación de una experiencia de aprendizaje positiva sobre los métodos ágiles y, por ende, carecen de la capacidad de motivar al estudiante durante el proceso de aprendizaje.	30	35	13	2	0	1.2	0.8
	E6. Las metodologías tradicionales de enseñanza no logran establecer una conexión genuina con los desafíos que podrían surgir en un entorno laboral real donde se utilicen métodos ágiles.	19	46	8	5	2	0.9	0.9
Juegos educativos	E7. El uso de otros recursos didácticos, como los juegos educativos, podría enriquecer el aprendizaje sobre los métodos ágiles.	75	5	0	0	0	1.9	0.2
	E8. Los juegos educativos que emulan desafíos reales en el campo de los métodos ágiles tienen la capacidad de fomentar el aprendizaje práctico de los estudiantes.	62	9	6	2	1	1.6	0.8
	E9. Mediante el uso de juegos se puede lograr una aproximación más efectiva para aplicar el conocimiento adquirido en el entorno académico al mundo real.	71	2	4	1	2	1.7	0.8

Ítem	Escala de respuestas					Calificación	
	TA	DA	MI	ED	TD	M	DS
E10. Los juegos educativos podrían contribuir a mejorar la adquisición de conocimientos sobre los métodos ágiles, de tal manera que sea posible adaptarse más rápidamente a las demandas actuales de la industria.	14	66	0	0	0	1.2	0.4
E11. Los juegos educativos tienen el potencial para complementar las metodologías tradicionales de enseñanza en tópicos altamente prácticos de la Ingeniería de Software, como lo son los métodos ágiles.	76	0	4	0	0	1.9	1.4

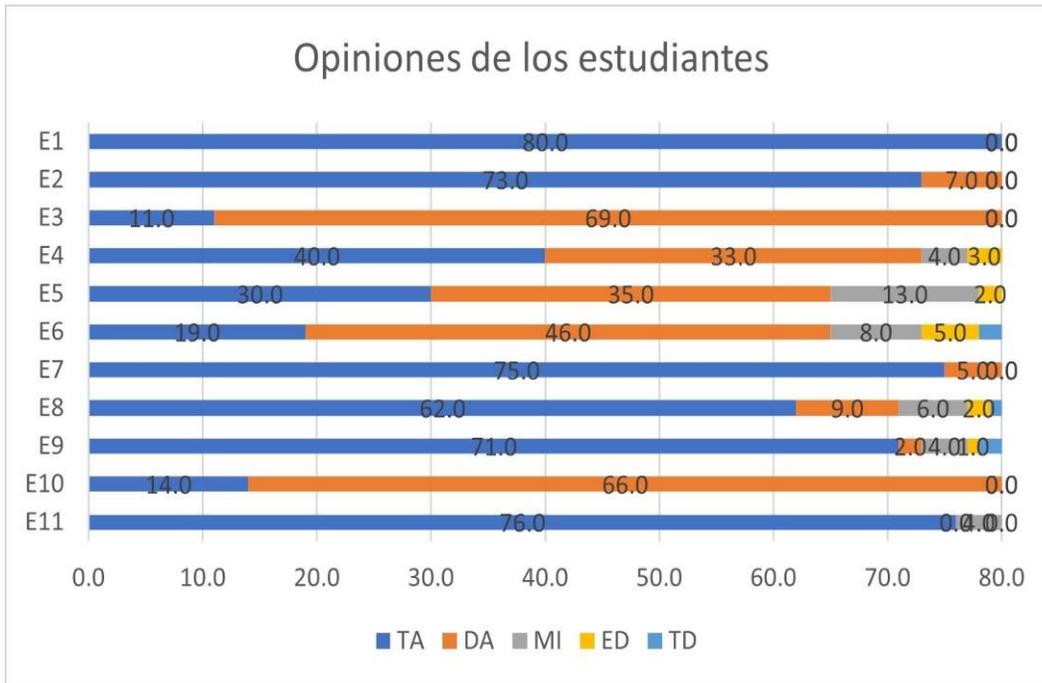
La Figura 3.5 muestra que, como resultado del análisis de las opiniones de los estudiantes sobre las metodologías tradicionales de enseñanza, se observó un panorama similar al reflejado por los comentarios de los profesores, puesto que el 92.3% de los estudiantes consideró que éstas no son adecuadas para la enseñanza de los métodos ágiles. A pesar de que no es un número representativo, tres estudiantes estuvieron de acuerdo que un enfoque tradicional podría funcionar para aprender lo necesario sobre un método ágil, mientras que seis estudiantes se mostraron indiferentes a la pregunta.



**Figura 3.5.** Opiniones de los estudiantes con relación a las metodologías tradicionales de enseñanza

La Figura 3.6 resume las opiniones expresadas por los estudiantes y, como se observa, el 100% consideró que una clase totalmente teórica es inadecuada para la enseñanza de los métodos ágiles. Es decir, los estudiantes consideran que una clase puramente teórica se vuelve monótona y se termina perdiendo el interés en el material presentado por el profesor (*E1. Las metodologías tradicionales de enseñanza que se basan exclusivamente en clases teóricas son monótonas y carentes de interés*,  $M=2.0$ ,  $DS=0.0$ ). Además, todos los participantes consideraron que este enfoque no facilita la adquisición de experiencia (*E2. Las metodologías tradicionales de enseñanza no facilitan la adquisición de experiencia práctica relacionada con la aplicación de conceptos teóricos sobre los métodos ágiles de desarrollo en situaciones reales*,  $M=1.9$ ,  $DS=0.3$ ) y ponen en riesgo el aprendizaje práctico durante el curso (*E3. Las metodologías tradicionales de enseñanza no son adecuadas para fomentar el aprendizaje práctico sobre los métodos ágiles*,  $M=1.1$ ,  $DS=0.3$ ). En contraste, el 19% de los estudiantes consideró que es posible que un enfoque tradicional de enseñanza contribuya a que se desempeñen correctamente en la industria (*E6. Las metodologías tradicionales de enseñanza no logran establecer una conexión genuina con los desafíos que podrían surgir en un entorno laboral*

real donde se utilicen métodos ágiles,  $M=0.9$ ,  $DS=0.9$ ). De hecho, este ítem reflejó la mayor discrepancia entre los estudiantes. Es importante mencionar que esta información no contradice la hipótesis establecida en la presente tesis, al contrario, se considera que la creación de un juego serio podría complementar aún más el entrenamiento de este 19% de estudiantes que están satisfechos con el enfoque tradicional y, por ende, generar en ellos una experiencia de aprendizaje más productiva.



**Figura 3.6.** Distribución de respuestas en el cuestionario aplicado a los estudiantes participantes

Ahora bien, con relación a la opinión de los estudiantes sobre el uso de juegos serios como soporte educativo en un curso sobre métodos ágiles, se obtuvo una pequeña variación a favor considerando las opiniones de sus profesores (véase Figura 3.7). Concretamente, el 100% de los estudiantes argumentó que los juegos serios podrían mejorar su aprendizaje sobre tales métodos (*E7. El uso de otros recursos didácticos, como los juegos educativos, podría enriquecer el aprendizaje sobre los métodos ágiles*,  $M=1.9$ ,  $DS=0.2$ ) y ayudarlos a adquirir experiencia que es directamente aplicable a su vida profesional (*E10. Los juegos educativos podrían contribuir a mejorar la adquisición de conocimientos sobre los métodos ágiles, de tal manera que sea posible adaptarse más rápidamente a las demandas actuales de la industria*,  $M=1.2$ ,  $DS=0.4$ ).

Por otro lado, el 95% de los estudiantes consideró que los juegos serios podrían utilizarse para mejorar el aprendizaje en áreas de la Ingeniería de Software que requieran de una práctica considerable (*E11. Los juegos educativos tienen el potencial para complementar las metodologías tradicionales de enseñanza en tópicos altamente prácticos de la Ingeniería de Software, como lo son los métodos ágiles*,  $M=1.9$ ,  $DS=1.4$ ), el restante 5% se mostró indiferente a la afirmación. Las mayores discrepancias entre los estudiantes se dieron para los ítems *E8* y *E9* puesto que el 11% de los estudiantes desconfió de que los juegos serios puedan emular escenarios reales de aprendizaje.

Finalmente, los resultados de la aplicación de los cuestionarios, tanto a profesores como estudiantes, justificaron, en el contexto de esta tesis, el desarrollo de un juego serio puesto que se determinó que existía disposición por parte de ambos grupos de participantes de usar un recurso educativo alternativo tanto para promover la enseñanza como el aprendizaje. Por lo tanto, se recolectó

información que permitió iniciar formalmente del desarrollo de las pre-fases establecidas en de Lope et al., (2015).



**Figura 3.7.** Opiniones de los estudiantes con relación a los juegos serios como recurso educativo

### 3.1.1.2. Análisis de planes de estudio y temarios para identificar los objetivos educativos iniciales

En segunda instancia, con el propósito de identificar las habilidades esenciales y los objetivos educativos que aborda un temario sobre entornos ágiles de desarrollo a nivel licenciatura, se realizó una revisión de los planes de estudio utilizados en el SUNE0. En este sentido, resultó fundamental examinar los planes de estudio de las universidades incorporadas a dicho sistema educativo que ofrecieran programas de Licenciatura en Informática o Ingeniería en Computación y que, además, incluyeran materias relacionadas con los métodos ágiles y el proceso de desarrollo de información para los usuarios. De esta manera, dicho análisis sirvió para identificar elementos comunes en los contenidos educativos enfocados a la enseñanza de tales métodos. A continuación, se presenta un resumen de los principales hallazgos.

- Las universidades en donde se imparte el programa de Ingeniería en Computación son la UTM, la UNISTMO (campus Tehuantepec), y la UNPA (campus Loma Bonita). El plan de estudios de dicho programa define, para el 5° semestre, un curso titulado “*Ingeniería de Software*” en el cual, de acuerdo con la documentación oficial, se abordan las diferentes fases que conforman un ciclo de vida de software. Sin embargo, el temario no especifica si es obligatorio que el profesor cubra un ciclo de vida tradicional o si puede, por iniciativa propia, darle una orientación al curso hacia el uso de los métodos ágiles. Por lo tanto, esta decisión se basa únicamente en el criterio del profesor. Además, también dependerá de su disposición el incluir o no algunos temas relacionados con la definición y el uso de un proceso para desarrollar la documentación necesaria para el usuario en un entorno ágil de trabajo. Por otro lado, los programas de las tres universidades incluyen un curso denominado “*Desarrollo ágil de software*” para estudiantes del 8° semestre. De acuerdo con su temario, dicho curso presenta a los estudiantes las principales características de tres métodos ágiles (i.e., programación extrema, Scrum, y *Lean*) con el objetivo de que éstos sean aplicados en el desarrollo de software. Aunado a lo anterior, se aborda la gestión ágil de proyectos de software haciendo hincapié en la definición de estrategias y generación de historias de usuario para la obtención

de requisitos “ágiles”; sin embargo, no se menciona algún método o proceso específico para desarrollar la documentación requerida por los usuarios de cualquiera de los tres métodos ágiles abordados en el curso. Por lo tanto, también queda a elección del profesor el introducir o no un estándar que contenga este conocimiento. Ambos cursos tienen una duración de 80 horas al semestre para las tres universidades.

- Las universidades en donde se imparte el programa de Licenciatura en Informática son la UNISTMO (campus Ixtepec), la UMAR (campus Puerto Escondido), la UNSIS, la UNSIJ, la UNCA y NovaUniversitas. El plan de estudios del programa establece que los estudiantes tomen dos cursos relacionados directamente con la Ingeniería de Software. El curso de “*Ingeniería de Software I*”, por ejemplo, se imparte en el 5° semestre con el fin de que los estudiantes se relacionen con las características de las fases de requisitos y análisis y diseño en un ciclo de vida de software, incluyendo a los métodos ágiles. Además, también se incluyen temas relacionados con el proceso de documentación requerida en el desarrollo de software tradicional como, por ejemplo, la creación y documentación de diagramas de casos de uso. El curso de “*Ingeniería de Software II*”, por otro lado, se debe impartir de forma seriada en 6° semestre con el principal objetivo de que los estudiantes complementen su formación sobre el área agregando temas específicos que se relacionan con las fases de implementación, pruebas, implantación, y mantenimiento del software. Aunado a esto, el temario define que debe incluirse material para explicar a los estudiantes la importancia de crear manuales de usuario, pero desde una perspectiva de producto y no de proceso. Desafortunadamente, no se observa una orientación hacia un enfoque ágil de desarrollo y, como se mencionó para el caso de la Ingeniería en Computación, dependerá del profesor agregar o no material relacionado con la documentación en entornos ágiles de desarrollo. Finalmente, se establece que en 8° semestre los estudiantes tomen el curso de “*Calidad de software*”, cuyo objetivo es que comprendan las métricas, normas y estándares de calidad necesarios para desarrollar software de manera eficiente. Tampoco se observa una orientación hacia los métodos ágiles, pero, a pesar de esto, se proporcionan conocimientos a los estudiantes sobre estándares del proceso de software, como ISO/IEC 15504 – *Software Process Improvement Capability Determination* e ISO/IEC 9126 – *Software Engineering - Product quality*. Este último aspecto resulta relevante para justificar el desarrollo de la presente tesis, puesto que se debe recordar que se pretende crear una propuesta que complemente la educación de los estudiantes universitarios sobre un estándar propio de los métodos ágiles. Los tres cursos tienen una duración de 80 horas por semestre cada uno.

Como se puede observar, resulta complicado discernir únicamente con los temarios, si los profesores que imparten cualquiera de los cinco cursos encontrados para las nueve instituciones académicas del SUNEО abordan a detalle las características de los métodos ágiles y, mejor aún, si se explora la importancia de generar documentación útil para el usuario. Dado que en gran medida depende del profesor titular el agregar o no una técnica, método o, incluso, la cantidad de trabajo práctico que los estudiantes deberán realizar para reafirmar los conceptos teóricos aprendidos en el aula, se entiende que los temarios son lineamientos curriculares básicos del curso que se adaptarán al enfoque que se pretenda impartir. Además, los programas de estudio son prácticamente idénticos para todas las universidades del mencionado sistema, por lo que fue posible identificar aspectos comunes que deberían considerarse si se planteara realizar una propuesta educativa conjunta. Las Tablas 5 y 6 sintetizan el consenso de la información relacionada con los cursos analizados.

**Tabla 5.** Análisis sobre el contenido de los temarios y lineamientos para la carrera de Ingeniería en Computación

<b>Universidades: UTM, UNISTMO (campus Tehuantepec), UNPA (campus Loma Bonita)</b>	
Cursos impartidos:	<ol style="list-style-type: none"> <li>1. Ingeniería de Software.</li> <li>2. Desarrollo ágil de software.</li> </ol>
Temas cubiertos (considerando los tópicos relacionados con la definición y uso de los métodos ágiles y el desarrollo de la documentación respectiva):	<ol style="list-style-type: none"> <li>1. Ciclos de vida del software.</li> <li>2. Administración de proyectos de software.</li> <li>3. Historia de los métodos ágiles.</li> <li>4. Características de los métodos ágiles.</li> <li>5. Métodos ágiles.</li> <li>6. Herramientas de software para el desarrollo ágil.</li> <li>7. Programación extrema (definición, principios básicos, roles, proceso de desarrollo, prácticas de programación).</li> <li>8. Scrum (definición, principios básicos, roles, proceso de desarrollo).</li> <li>9. <i>Lean</i> (definición, principios básicos, roles, proceso de desarrollo).</li> <li>10. Gestión ágil de proyectos (requerimientos ágiles, planificación y estimación ágil, calidad en entornos ágiles, retrospectivas ágiles y mejora continua).</li> </ol>
Actividades de enseñanza y aprendizaje:	<p>Para el curso de “Ingeniería de Software” se sugiere:</p> <ol style="list-style-type: none"> <li>1. Revisión bibliográfica del tema por los estudiantes en libros.</li> <li>2. Aplicación de prácticas adecuadas a cada tema de las áreas de conocimiento de la Ingeniería de Software, usando para ello herramientas actuales de cada área.</li> </ol> <p>Para el curso de “Desarrollo ágil de software” se sugiere:</p> <ol style="list-style-type: none"> <li>1. Desarrollar actividades de aprendizaje que propicien la aplicación de los conceptos, modelos y metodologías que se van aprendiendo en el desarrollo de la asignatura.</li> </ol>
Criterios de evaluación:	Para aprobar el curso se realizarán tres evaluaciones parciales (50%) y una evaluación final (50%). Para cada evaluación se realizará un examen y se evaluarán tareas y proyectos. El examen tendrá un valor mínimo de 50% y las tareas y proyectos un valor máximo de 50%.
Método, modelo o estándar incluido para soportar la enseñanza del desarrollo de documentación en entornos ágiles:	Los temarios no especifican que técnicas, métodos, modelos o estándares del proceso de software es necesario abordar a lo largo del curso.

**Tabla 6.** Análisis sobre el contenido de los temarios y lineamientos para la carrera de Licenciatura en Informática

<b>Universidades: UNISTMO (campus Ixtepec), UMAR (campus Puerto Escondido), UNSIS, UNSIJ, UNCA, NovaUniversitas</b>	
Cursos impartidos:	<ol style="list-style-type: none"> <li>1. Ingeniería de Software I.</li> <li>2. Ingeniería de Software II.</li> <li>3. Calidad de software.</li> </ol>
Temas cubiertos (considerando los tópicos relacionados con la definición y uso de los métodos ágiles y el desarrollo de la documentación respectiva):	<ol style="list-style-type: none"> <li>1. Modelos de procesos de software (modelos ágiles).</li> <li>2. Especificación de requerimientos (documentación de casos de uso, documentación de requerimientos).</li> <li>3. Verificación y validación (generación del plan de pruebas).</li> <li>4. Elaboración de manuales.</li> <li>5. Modelos de empresa y personal para desarrollo de software (ISO/IEC 15504 – Evaluación de procesos de software).</li> <li>6. Métricas de software (ISO/IEC 9126 - Métricas de calidad).</li> </ol>
Actividades de enseñanza y aprendizaje:	<p>Para el curso de “Ingeniería de Software I” se sugiere:</p> <ol style="list-style-type: none"> <li>1. Hacer énfasis en la segunda unidad en el modelo de proceso unificado.</li> <li>2. Plantear un proyecto de software en la tercera unidad, para que se seleccione un modelo de proceso adecuado. Este proyecto se</li> </ol>

Universidades: UNISTMO (campus Ixtepec), UMAR (campus Puerto Escondido), UNSIS, UNSIJ, UNCA, NovaUniversitas	
	<p>desarrollará durante el curso y deberá concluirse en la materia “Ingeniería de Software II”.</p> <p>Para el curso de “Ingeniería de Software II” se sugiere:</p> <ol style="list-style-type: none"> <li>1. Retomar el proyecto con los avances respectivos del análisis y diseño planteados en la materia prerequisite. El proyecto deberá concluirse al finalizar el curso.</li> </ol> <p>Para el curso de “Calidad de software” se sugiere:</p> <ol style="list-style-type: none"> <li>1. Realizar investigación sobre los temas actuales de las áreas en las unidades de aprendizaje.</li> <li>2. Para cada unidad de aprendizaje se diseñarán, desarrollarán o utilizarán programas especializados para que el alumno practique lo expuesto en clases.</li> </ol>
Criterios de evaluación:	<p>Se realizan tres evaluaciones parciales y una evaluación ordinaria final de la asignatura. Para las evaluaciones parciales, se deberá realizar un examen escrito y se podrá complementar la evaluación con exámenes prácticos, avances de proyectos, tareas, investigaciones y otras actividades académicas previamente aprobadas de acuerdo con la normatividad Universitaria. Queda a criterio del profesor la ponderación de todas las actividades.</p> <p>Para la evaluación ordinaria final, se deberá realizar un examen escrito y se podrá complementar la evaluación con proyectos, exposiciones, tareas e investigaciones realizadas a lo largo del semestre. Queda a criterio del profesor la ponderación de todas las actividades.</p> <p>Para la calificación final de la asignatura, se establece la ponderación de las evaluaciones parciales y ordinaria final con base en la normatividad de la Universidad.</p>
Método, modelo o estándar incluido para soportar la enseñanza del desarrollo de documentación en entornos ágiles:	Los temarios no especifican que técnicas, métodos, modelos o estándares del proceso de software es necesario abordar a lo largo del curso.

Considerando la información de ambas tablas, se llegó a los siguientes hallazgos:

- En el curso de “*Ingeniería de Software*”, impartido en la UTM, UNISTMO (campus Tehuantepec) y UNPA (campus Loma Bonita), se presenta al proceso de desarrollo de software desde un enfoque tradicional y no se incluye material relacionado con un enfoque ágil. No obstante, dado que existe la flexibilidad en el proceso de enseñanza y aprendizaje, el profesor tiene la libertad de realizar actividades que propicien la aplicación de conceptos teóricos que se aprendan durante el curso. Es decir, se podría, si él lo decide, incorporar contenido sobre algún método ágil que complemente el enfoque tradicional de desarrollo y usar como actividad práctica el juego serio propuesto en esta tesis.
- En el curso de “*Desarrollo ágil de software*”, impartido también en la UTM, UNISTMO (campus Tehuantepec) y UNPA (campus Loma Bonita), se observó una buena aproximación a las principales características de los métodos ágiles. De hecho, la revisión de tres de los principales métodos usados en la industria moderna de software es evidencia de que los estudiantes recibirán conocimiento actual que podrán aplicar en su vida profesional. Otro aspecto positivo es que el temario hace alusión a aspectos básicos de documentación que bien podrían fortalecerse con la incorporación del estándar ISO/IEC/IEEE 26515:2018, con el fin de que los estudiantes comprendan la importancia de la documentación en entornos ágiles de desarrollo a través de, por ejemplo, el juego serio propuesto en esta tesis.
- En los cursos de “*Ingeniería de Software I*” e “*Ingeniería de Software II*” impartidos en la UNISTMO (campus Ixtepec), UMAR (campus Puerto Escondido), UNSIS, UNSIJ, UNCA y NovaUniversitas, se abordan tópicos específicos que exponen la necesidad de documentar las

fases de un ciclo de vida de software. Es verdad que siguiendo este enfoque es posible concientizar a los estudiantes sobre la importancia de documentar el proceso de desarrollo de software; sin embargo, los tópicos explorados no se relacionan con las técnicas requeridas en entornos que utilizan métodos ágiles de desarrollo.

- En el curso de “*Calidad de software*”, impartido también en las seis anteriores universidades, se introducen los estándares ISO/IEC 15504 e ISO/IEC 9126, algo que no se observó en el resto de los cursos. Por lo tanto, se trata de un aspecto bastante positivo en la educación de un futuro profesionalista del desarrollo de software. Desafortunadamente, el temario no especifica concretamente qué actividades de aprendizaje debería realizar el estudiante para comprender los conceptos fundamentales de tales estándares, dado que se trata de documentos con una terminología demasiado compleja y técnica. No obstante, al indicar que se “...utilizarán programas especializados para que el alumno practique lo expuesto en clases”, se entiende que el profesor decidirá o no el incorporar herramientas que reafirmen el aprendizaje de los estudiantes.
- El programa de Ingeniería en Computación establece la necesidad de realizar actividades prácticas que sean adecuadas a cada tema y, además, que requieran el uso de herramientas actuales para cada fase de la Ingeniería de Software, por lo que se le brinda al profesor la oportunidad de introducir herramientas educativas de soporte para mejorar el proceso de enseñanza/aprendizaje. Sin embargo, el programa de Licenciatura en Informática no da cabida a la incorporación de este tipo de actividades y, por ende, un profesor se ve bastante limitado a realizar actividades adicionales para reafirmar los conceptos teóricos aprendidos en el aula.
- Ambos programas educativos establecen que se deberán realizar evaluaciones parciales mediante la aplicación de exámenes teóricos, y que además tales evaluaciones se deberán complementar con avances periódicos de un proyecto que sea asignado como parte práctica del curso. Teniendo esto en consideración, es evidente que los estudiantes están relacionados con el desarrollo de actividades prácticas y que se comprende la importancia de reafirmar la teoría con la práctica. Razón por la cual se considera que el juego serio propuesto en esta tesis podría ser bien recibido tanto por estudiantes como profesores.
- Finalmente, es importante enfatizar en que no se observó la incorporación de material relacionado con el estándar ISO/IEC/IEEE 26515:2018 en ninguno de los cursos analizados. No obstante, se detectó la posibilidad de agregar material y actividades al curso de “*Diseño ágil de software*” que bien podría complementar la formación de los estudiantes en el uso y gestión de los métodos ágiles.

Es evidente que el propósito de esta tesis no se enfocó en la creación de un nuevo curso para las universidades del SUNEI, puesto que la actualización de los planes de estudio y contenidos temáticos en estas instituciones sigue procedimientos bien definidos que están a cargo de expertos en la materia, siendo este grupo de especialistas el único autorizado para hacer las correspondientes modificaciones a los cursos. Por lo contrario, hasta esta parte del documento de tesis se ha resaltado la importancia de desarrollar un soporte educativo para la enseñanza de los métodos ágiles y el desarrollo de información útil para sus usuarios. A partir de esta perspectiva, se identificaron los siguientes objetivos educativos que debieron considerarse en el diseño del juego serio presentado en esta tesis:

- Generar una dinámica de aprendizaje basada en el juego que complemente la educación de los estudiantes sobre los métodos ágiles.
- Abordar los procesos y actividades del estándar ISO/IEC/IEEE 26515:2018, con el fin de que los estudiantes comprendan los conceptos fundamentales sobre la documentación en entornos ágiles de desarrollo.

- Relacionar a los estudiantes con las técnicas utilizadas para desarrollar la distinta documentación requerida por el usuario, así como con los diferentes tipos de documentos requeridos en un entorno ágil de desarrollo.
- Fortalecer el aprendizaje de los conceptos teóricos a través de sesiones de juego.

### 3.1.1.3. Identificación de las competencias básicas

Considerando la información anterior, y dado que no se pretendió crear un nuevo curso sobre métodos ágiles, se decidió analizar tanto los Lineamientos Curriculares para Programas de Licenciatura en Ingeniería de Software 2014<sup>3</sup> (SE 2014, por sus siglas en inglés) como el Modelo de Competencias de la Ingeniería de Software<sup>4</sup> (SWECOM, por sus siglas en inglés), con el objetivo de identificar aspectos teóricos y prácticos sugeridos por organismos especialistas como la ACM y el IEEE para la enseñanza de tales métodos a nivel licenciatura.

En este sentido, el documento de SE 2014 identifica a 10 áreas de conocimiento que conforman un cuerpo de Conocimientos sobre la Educación de la Ingeniería de Software (SEEK, por sus siglas en inglés), las cuales son: conceptos básicos de computación, fundamentos matemáticos y de ingeniería, práctica profesional, modelado y análisis de software, análisis y especificación de requisitos, diseño de software, verificación y validación de software, proceso de software, calidad de software, y seguridad. Así, SE 2014 establecen que la introducción de los estudiantes universitarios a los métodos ágiles se dé en el área de conocimiento denominada “Proceso de software” (véase Tabla 7). Los tópicos de esta área de conocimiento están divididos en cinco unidades (resaltadas con negrilla en la Tabla 7) que involucran diferentes conocimientos tanto teóricos como prácticos: PRO.con, PRO.imp, PRO.pp, PRO.cm y PRO.evo. Por lo tanto, de acuerdo con SE2014, los métodos ágiles son abordados específicamente en los tópicos PRO.con.7, PRO.imp.2, PRO.pp.1, PRO.pp.2, y PRO.pp.6, teniendo en consideración que, durante la unidad PRO.pp, se sugiere la impartición tanto de la gestión de proyectos de software con un enfoque tradicional como con los métodos ágiles. SE 2014 hacen hincapié en que es importante mantener separadas estas dos alternativas, su terminología y sus fortalezas y debilidades.

**Tabla 7.** Tópicos relacionados con el área de conocimiento “Proceso de software” (adaptada de ACM-IEEE, 2014)

Nivel de cobertura	Tópico	Nivel cognitivo	Relevancia del tópico
<b>PRO.con</b>	<b>Conceptos de proceso</b>		
PRO.con.1	Temas y terminología.	K	E
PRO.con.2	Infraestructura de procesos de Ingeniería de Software (e.g., personal, herramientas, capacitación).	K	E
PRO.con.3	Modelado y especificación de procesos de software.	C	E
PRO.con.4	Medición y análisis de procesos de software.	C	E
PRO.con.5	Mejora de los procesos de Ingeniería de Software (e.g., individual, equipo, organización).	C	E

<sup>3</sup> SE 2014 fueron creados 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 o actualizar programas de licenciatura sobre Ciencias de la Computación, Ingeniería en Computación, Ingeniería de Software, y Sistemas de Información (IEEE-ACM, 2015).

<sup>4</sup> SWECOM fue creado por el IEEE y la Sociedad de Computación del IEEE para especificar las competencias (i.e., conocimientos y habilidades) de los ingenieros de software que participan en el desarrollo y modificación de sistemas intensivos de software (IEEE-SC, 2014).

Nivel de cobertura	Tópico	Nivel cognitivo	Relevancia del tópico
PRO.con.6	Análisis y control de la calidad (e.g., prevención de defectos, revisión de procesos, métricas de calidad, y análisis de las causas raíz de defectos críticos para mejorar los procesos y las prácticas).	C	E
PRO.con.7	Modelos de ciclos de vida de la Ingeniería de Software (e.g., basado en planes, incremental, iterativo, ágil).		D
<b>PRO.imp</b>	<b>Implementación del proceso</b>		
PRO.imp.1	Niveles de definición de procesos (e.g., organización, proyecto, equipo, individual).	K	E
PRO.imp.2	Características del modelo de ciclo de vida (e.g., basado en planes, incremental, iterativo, ágil).	C	E
PRO.imp.3	Proceso individual de software (i.e., modelo, definición, medición, análisis y mejora).	A	E
PRO.imp.4	Proceso de equipo de software (i.e., modelo, definición, organización, medición, análisis y mejora).	A	E
PRO.imp.5	Implementación del proceso de software en el contexto de la Ingeniería de Software.	K	E
PRO.imp.6	Adaptación del proceso.	K	E
PRO.imp.7	Efecto de factores externos (e.g., requisitos contractuales y legales, estándares y prácticas de adquisición) en el proceso de software.	K	E
<b>PRO.pp</b>	<b>Planificación y monitorización de proyectos</b>		<b>E</b>
PRO.pp.1	Gestión de requisitos (e.g., producto backlog, prioridades, dependencias y cambios).	A	E
PRO.pp.2	Estimación de esfuerzo (e.g., uso de datos históricos y técnicas de estimación basadas en contexto).	A	E
PRO.pp.3	Estructura de desglose de trabajo y calendarización de tareas.	A	E
PRO.pp.4	Asignación de recursos.	C	E
PRO.pp.5	Gestión de riesgos (e.g., identificación, mitigación, corrección y seguimiento de estado).	A	E
PRO.pp.6	Métricas y técnicas de seguimiento de proyectos (e.g., valor ganado, velocidad, gráficos de quemado, seguimiento de defectos, gestión de la deuda técnica).	A	E
PRO.pp.7	Autogestión del equipo (e.g., seguimiento del proceso, asignación dinámica de la carga de trabajo, respuesta a problemas emergentes).	A	E
<b>PRO.cm</b>	<b>Gestión de la configuración de software</b>		
PRO.cm.1	Control de revisión.	A	E
PRO.cm.2	Gestión de versiones.	C	E
PRO.cm.3	Herramientas de gestión de la configuración.	C	E
PRO.cm.4	Construcción de procesos y herramientas, incluyendo a las pruebas automatizadas y la integración continua.	A	E
PRO.cm.5	Procesos de la gestión de configuración de software.	K	E
PRO.cm.6	Problemas de mantenimiento.	K	E
PRO.cm.7	Distribución y respaldo.		D
<b>PRO.evo</b>	<b>Procesos y actividades de evolución</b>		
PRO.evo.1	Conceptos básicos de evolución y mantenimiento.	K	E
PRO.evo.2	Trabajo con sistemas heredados.	K	E
PRO.evo.3	Refactorización.	C	E

**K** = Conocimiento, **C** = Comprensión, **A** = Aplicación, **E** = Esencial, **D** = Deseable

Además, para cada tópico incluido en cada una de las áreas de conocimiento, SE 2014 designan un nivel de la Taxonomía de Bloom para indicar la capacidad que debe poseer un graduado sobre cada tópico (K – Conocimiento, C – Comprensión, A – Aplicación) y la relevancia del tema para indicar si

el tema es esencial (E) o deseable (D) a nivel licenciatura. En este sentido, los tópicos de SE 2014 relacionados con los métodos ágiles requieren niveles cognitivos para conocimiento y aplicación con una relevancia, en su mayoría, esencial.

Por otro lado, en el SWECOM se identifican áreas, habilidades y competencias que se dividen en dos grandes grupos: áreas y habilidades del ciclo de vida de la Ingeniería de Software (requisitos de software, diseño de software, construcción de software, pruebas de software, mantenimiento de software) y áreas de competencias transversales de la Ingeniería de Software (proceso y ciclo de vida de software, ingeniería de sistemas de software, calidad de software, seguridad de software, protección de software, gestión de configuración de software, medición de software, interacción humano-computadora). Los métodos ágiles, que son el tópico de interés en esta tesis, se encuentran en el área transversal denominada “Proceso y ciclo de vida de software”, específicamente en la habilidad de “Implementación del ciclo de vida del desarrollo de software”, puesto que el resto de las habilidades no están incluidas en los planes de estudio a nivel licenciatura (véase Tabla 8).

**Tabla 8.** Lineamientos y recomendaciones del SWECOM para la enseñanza de los métodos ágiles (resumida de IEEE-SC, 2014)

Área de la Ingeniería de Software	Conjunto de habilidades	Actividades prácticas sugeridas
Proceso y ciclo de vida de software	1. Implementación del ciclo de vida del desarrollo de software.	1. Determinar uno o más modelos de ciclo de vida para una organización (como modelos en cascada, en espiral, en V, incrementales, de madurez, o ágiles). 2. Seleccionar un proceso de software de equipo (basado en un plan, adaptativo). 3. Liderar un pequeño equipo en la ejecución de alguna parte de un modelo de proceso del ciclo de vida (como el diseño de software). 4. Llevar a cabo actividades de proceso especificadas en un guion de modelo de proceso de ciclo de vida.
	2. Definición y adaptación de procesos.	5. Definir procesos de software para un equipo de proyecto o para una actividad de Ingeniería de Software (como la ingeniería de requisitos). 6. Adaptar un proceso de software definido a las necesidades de un equipo de proyecto o actividad de Ingeniería de Software. 7. Interpretar y adaptar un proceso de software a las responsabilidades y tareas de los procesos individuales. 8. Liderar la definición y adaptación de procesos de software en toda la organización.
	3. Implementación y gestión de procesos.	9. Implementar y ejecutar procesos de software. 10. Proporcionar orientación y asesoramiento a los equipos de software sobre cómo implementar y gestionar procesos de software. 11. Servir como miembro de un grupo de procesos de Ingeniería de Software.
	4. Evaluación y mejora de procesos.	12. Recopilar datos para la evaluación de un proceso de software. 13. Analizar datos de evaluación de procesos e implementar mejoras en los procesos de software del equipo.

		14. Utilizar información e informes de evaluación para mejorar los procesos de software.
--	--	--

Para determinar el nivel de competencia que un futuro profesional de la industria de software debería tener sobre el área transversal mostrada en la tabla anterior, el SWECOM establece los siguientes cinco niveles de competencia: técnico (una persona que se limita a seguir instrucciones), practicante de nivel inicial (una persona que ayuda en la realización de alguna actividad o bien realiza la actividad pero bajo supervisión), practicante (una persona que realizar una o más actividades sin la necesidad de supervisión), líder técnico (una persona que lidera individuos y equipos en el desempeño de actividades) e ingeniero sénior de software (una persona que es capaz de modificar métodos y herramientas existentes con el objetivo de crear otras nuevas). De hecho, algunos ingenieros de software sénior pueden ser reconocidos como expertos de la industria que contribuyen a dar forma y mejorar la profesión de la Ingeniería de Software.

De acuerdo con el SWECOM, algunas organizaciones fusionan los niveles de técnico y practicante de nivel inicial en uno solo y, por esta razón, podría considerarse que son las competencias con las que un graduado de nivel licenciatura debería contar puesto que sería su primer contacto con la industria. Sin embargo, un estudiante de licenciatura no puede ser un practicante de nivel inicial puesto que carece de experiencia práctica llevada a cabo en algún entorno real de trabajo, razón por la cual, en el contexto de esta tesis, su nivel adecuado es el de técnico. Por otro lado, es importante mencionar que el SWECOM no aborda competencias relacionadas con el uso de herramientas ni el cumplimiento de estándares del proceso de software prescritos para realizar las actividades, puesto que argumenta que éstos son específicos de, ya sea, la orientación que se dé a un curso o bien de la necesidad específica de una organización o proyecto.

Aunado a lo anterior, el modelo también define una notación específica para identificar el nivel de involucramiento de las personas en las actividades de la Ingeniería de Software. Dicha notación establece que una persona puede Seguir (S), Asistir (A), Participar (P), Dirigir (D) y Crear (C) durante la realización de las actividades concretas relacionadas con el área de “Proceso y ciclo de vida de software” y la habilidad de “Implementación del ciclo de vida del desarrollo de software” (véase Tabla 9).

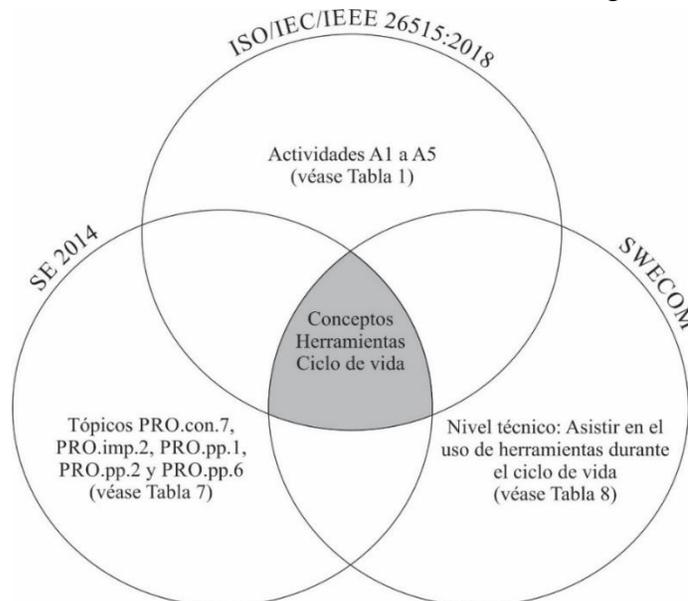
**Tabla 9.** Actividades y niveles de competencia para desarrollar habilidades con los métodos ágiles (resumida de IEEE-SC, 2014)

Habilidad	Niveles de competencia				
	Técnico	Practicante de nivel inicial	Practicante	Líder técnico	Ingeniero sénior de software
Implementación del ciclo de vida del desarrollo de software	1. Proporcionar asistencia en la instalación y uso de herramientas apropiadas para el modelo de ciclo de vida designado para el proyecto. (S/A).	1.Llevar a cabo las actividades del proceso especificadas en un guion del modelo de ciclo de vida. (A/P).	1. Liderar a un pequeño equipo en la ejecución de alguna parte de un modelo de ciclo de vida (como Scrum). (P/D).	1. Seleccionar un modelo de ciclo de vida para un equipo de software. (D). 2. Ayudar en la selección de un modelo de ciclo de vida para una organización o departamento. (P/D).	1. Determinar la necesidad y seleccionar o desarrollar un modelo de ciclo de vida para una organización. (C). 2. Ayudar en la selección de un modelo de ciclo de vida para una organización o departamento. (P/D).

Habilidad	Niveles de competencia				
	Técnico	Practicante de nivel inicial	Practicante	Líder técnico	Ingeniero sénior de software
					3. Seleccionar modelos de procesos para un departamento u organización. (C). 4. Asesorar sobre infraestructura de procesos, capacitación y herramientas. (C).

C = Crear, D = Dirigir, P = Participar, A = Asistir, S = Seguir

De la tabla anterior se puede concluir que, de acuerdo con el nivel de competencia de técnico, un estudiante de licenciatura debe adquirir experiencia para seguir instrucciones o, si es posible, asistir en la instalación y uso apropiado de herramientas que faciliten la ejecución del modelo de ciclo de vida escogido para el desarrollo de un proyecto de software. Es decir, se espera que un estudiante por lo menos aprenda a reconocer un conjunto de herramientas básicas que puedan facilitar el desarrollo de las actividades del modelo de ciclo de vida designado en la organización (e.g., incremental, iterativo, Scrum). Considerando la información anterior, la Figura 3.8 muestra las coincidencias existentes entre las sugerencias recomendadas por SE 2014 y SWECOM con respecto a la enseñanza de los métodos ágiles a nivel licenciatura, y las actividades del estándar ISO/IEC/IEEE 26515:2018 para el desarrollo de la documentación de los usuarios en un entorno ágil de desarrollo de software.



**Figura 3.8.** Aspectos comunes entre ISO/IEC/IEEE 26515:2018, SE 2014 y SWECOM

SE 2014 y SWECOM, ambas propuestas educativas, coinciden en que la formación de los estudiantes a nivel licenciatura debería incluir conocimientos relacionados con los conceptos fundamentales de un método ágil, las peculiaridades de estos modelos de ciclo de vida, el uso de

herramientas para mejorar su efectividad, y la realización de proyectos prácticos. Aspectos que únicamente coinciden con las actividades A1 a A5 del proceso de “Gestionar el desarrollo de la información” definido por el estándar ISO/IEC/IEEE 26515:2018. En este sentido, es necesario enfatizar en que, a diferencia de SE 2014 y SWECOM, dicho estándar no es una propuesta educativa, y que, por lo tanto, reúne una colección de actividades para definir de manera concreta el proceso de documentación en entornos ágiles de la industria. Por lo tanto, el estándar ISO/IEC/IEEE 26515:2018 representa el conocimiento que un profesional debería dominar para hacer su correcta implementación en un entorno ágil real considerando que “*Se supone que los usuarios de este documento tienen experiencia o conocimiento general sobre la información para usuarios (tradicionalmente llamada “documentación de usuario”) y los procesos ágiles*” (ISO/IEC/IEEE, 2018, pp. 1). Así pues, sería recomendable que los estudiantes de nivel licenciatura se centraran en comprender los conceptos fundamentales del estándar con el fin de, si se diera el caso, facilitar su implementación en su vida profesional.

### 3.1.2. Pre-2: Diseño del tipo de juego

Con el objetivo de cubrir los retos educativos, las competencias básicas y los objetivos educativos anteriores se consideró el diseño de un tablero digital siguiendo la idea del juego tradicional de mesa llamado “Serpientes y escaleras”. Dicho juego fue creado originalmente en la India con el propósito de promover la religión entre los niños al enseñarlos a diferenciar entre las cosas buenas (escaleras que representan buenas virtudes) y las malas (serpientes que representan vicios y mañas). Ortiz-Ramírez y Bravo-Agapito (2019) argumentaron que, desde la perspectiva de un contexto educativo, este tipo de juego presenta los siguientes beneficios:

- Ofrece momentos de diversión y recreación que permiten que los estudiantes aprendan y mejoren su capacidad de análisis, ya que tendrán que saber en qué dirección y por cuál escalera moverse para no perder. Asimismo, les ayuda a poner en práctica su concentración y, de esta manera, mejoran su memoria al retener más la información.
- Permite que los estudiantes aprendan a superar obstáculos que se les presenten en el camino. Para esto deben tomar una decisión y cada vez que lo hagan tendrán que reconocer las consecuencias que ésta traiga consigo si se equivocaron al elegir.
- Fomenta la competencia puesto que requiere la presencia de más de un jugador. Por lo tanto, los estudiantes no solamente ponen en práctica sus habilidades de sociabilización y empatía, sino que son motivados a realizar un mejor esfuerzo en cada partida.

Así, bajo la dinámica de este juego se fomenta la reflexión de los estudiantes con el objetivo de fortalecer, de forma lúdica, sus conocimientos sobre un tópico específico. En este sentido, el juego propuesto lleva el nombre de “*Serpientes y escaleras: Un juego de soporte al aprendizaje del estándar ISO/IEC/IEEE 26515:2018*”, en el que el estudiante debe alcanzar la última celda de un tablero digital dependiendo el resultado de tirar un dado virtual y contestando preguntas relacionadas con el estándar ISO/IEC/IEEE 26515:2018. Como es la naturaleza del juego, algunas celdas tienen serpientes que harán retroceder al estudiante, y otras, escaleras, que le permitirán avanzar más rápidamente por el tablero.

#### 3.1.2.1. Género y plataforma del juego

Considerando las clasificaciones propuestas por Arsenault (2009) y Konzack (2015), se trata de un juego digital de tablero que fue diseñado bajo los géneros de trivia y aventura. De acuerdo con García et al. (2020), los juegos de trivia ponen a prueba los conocimientos de los estudiantes sobre

temas específicos o sus habilidades para resolver desafíos respondiendo preguntas que les permiten obtener algún tipo de recompensa; mientras que los juegos de aventura implican más narrativa, ya que el estudiante normalmente desempeña un papel en una historia de fantasía.

En este sentido, la versión del juego de “Serpientes y escaleras” propuesta en esta tesis plantea una historia de fantasía que se desarrolla en una isla remota en la cual es común que sus habitantes realicen competencias para determinar quién es el más veloz. El jugador, o jugadores principales, deberán competir para saber quién llega a la meta en primer lugar y se hace del premio y reconocimiento de los habitantes de la isla. Sin embargo, en el camino se encontrarán atajos y trampas que se podrán librar al contestar correctamente tres de cinco preguntas, al menos, para subir por un atajo o evitar caer en una trampa que haga retroceder al jugador. El Anexo B de esta tesis muestra la información relevante sobre la trivia diseñada para el juego.

Finalmente, el juego digital se diseñó para ser usado en computadoras que cuenten con el sistema operativo Windows de Microsoft® en sus versiones 8, 9, 10 u 11.

### **3.1.2.2. Propósito y público objetivo**

Como se ha mencionado anteriormente, el propósito de esta versión digital del juego de “Serpientes y escaleras” es brindar a estudiantes y profesores universitarios una herramienta educativa que sirva de apoyo para la enseñanza/aprendizaje de los conceptos fundamentales del estándar ISO/IEC/IEEE 26515:2018. De esta manera, se buscó crear una herramienta divertida de soporte a cursos tradicionales que facilite a los estudiantes su acercamiento con las técnicas y procedimientos para desarrollar correctamente la documentación que es requerida por los usuarios en un entorno ágil de desarrollo de software.

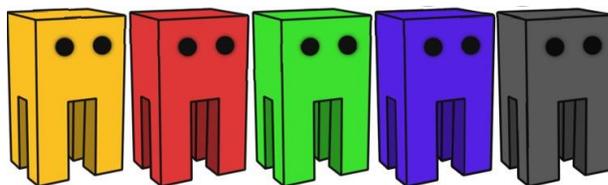
Teniendo esto en consideración, el público objetivo del videojuego son los estudiantes universitarios que están matriculados en programas académicos de licenciatura relacionados con las Ciencias de la Computación, específicamente aquellos que están cursando asignaturas relacionadas con la Ingeniería de Software y los métodos ágiles.

### **3.1.2.3. Área de aplicación**

El juego propuesto en esta tesis se ha creado considerando el contexto educativo de la Ingeniería de Software a nivel licenciatura, específicamente en el entrenamiento que los estudiantes deben recibir sobre la implementación, uso y modificación de los métodos ágiles. Como se observó en el análisis presentado en toda la sección 3.1.1 de este documento, los métodos ágiles son considerados en los planes actuales de estudio del SUNEО, por lo que la idea de fortalecer o complementar un método tradicional de enseñanza resulta lógica. Sin embargo, como se demostró con los argumentos expuestos en el Capítulo 2, desafortunadamente no existen juegos serios que aborden los conceptos fundamentales del estándar ISO/IEC/IEEE 26515:2018.

### **3.1.3. Pre-3: Diseño inicial de la historia y los personajes**

La historia se desarrolla en la isla mágica de Cubilandia. Este mundo ficticio es habitado por los “*cubitantes*”, seres pacíficos que disfrutaban de la naturaleza y el mar. Entusiasmados por destacar en la isla, estos personajes organizan competencias con mucha frecuencia con la intención de encontrar al habitante más veloz (véase Figura 3.9).



**Figura 3.9.** Personajes principales del juego

La competencia implica que los *cubitantes* recorran caminos dentro de la isla que ponen a prueba sus habilidades tanto físicas como mentales, puesto que deben pasar a través de distintos pasadizos que, de no estar atentos, dificultarán su recorrido. En este sentido, a lo largo del camino los competidores se encuentran escaleras que sirven de atajos para concluir el recorrido más rápidamente; sin embargo, para usar dichas escaleras los *cubitantes* deben responder correctamente al menos tres de cinco preguntas que les hará el guardián del camino. Si no se logra acertar la cantidad mínima requerida de preguntas, los *cubitantes* deberán avanzar por el camino sin usar los atajos, retrasando de esta manera el tiempo de culminación del recorrido. De manera similar, a lo largo del camino hay trampas peligrosas con serpientes que los *cubitantes* deberán librar para no retrasar la llegada a la meta. Si un *cubitante* llegará a caer en estas trampas, deberá también responder correctamente tres de cinco preguntas para no retroceder en su camino y continuar el recorrido desde ese mismo punto. En caso de que no se obtengan mínimamente los tres aciertos requeridos, el *cubitante* deberá retroceder y avanzar hacia la salida desde este nuevo punto. La aventura termina cuando todos los jugadores, a excepción de uno, logren llegar a la meta. Con el objetivo de encontrar al *cubitante* más veloz, los habitantes de Cubilandia han creado mapas con rutas que tienen diferentes niveles de dificultad, por lo que el participante ganador deberá mostrar un buen dominio y entendimiento del estándar.

### 3.1.3.1. Flujo del juego

Cuando el estudiante entra al juego visualiza la pantalla de inicio (véase Figura 3.10) con las siguientes opciones: ver las instrucciones del juego, iniciar el juego, salir del juego, y silenciar/activar el sonido.



**Figura 3.10.** Pantalla inicial del juego

Una vez que éste listo para iniciar deberá dar clic en el botón correspondiente para iniciar una sesión de juego.

-  Iniciar el juego.
-  Instrucciones del juego.
-  Salir del juego.
-  Silenciar/activar sonido.

Posteriormente, el estudiante debe elegir el nivel de dificultad con el que desea jugar (véase Figura 3.11).



**Figura 3.11.** Selección del nivel de dificultad del juego

Los niveles que maneja el juego corresponden con los mapas creados por los habitantes de Cubilandia y se describen de la siguiente manera:

- **Fácil:** En este nivel, el estudiante deberá responder preguntas dicotómicas con opciones de respuesta de verdadero o falso.
- **Medio:** En este nivel, el estudiante deberá responder preguntas de opción múltiple para las que elegirá la respuesta correcta entre tres distintas opciones.
- **Difícil:** En este nivel, el estudiante deberá introducir mediante el teclado su respuesta a la pregunta que se le presente. Las respuestas en este nivel siempre serán de una palabra de longitud. Como una ayuda para el jugador, se le mostrará la primera letra y la longitud de caracteres de la palabra respuesta.

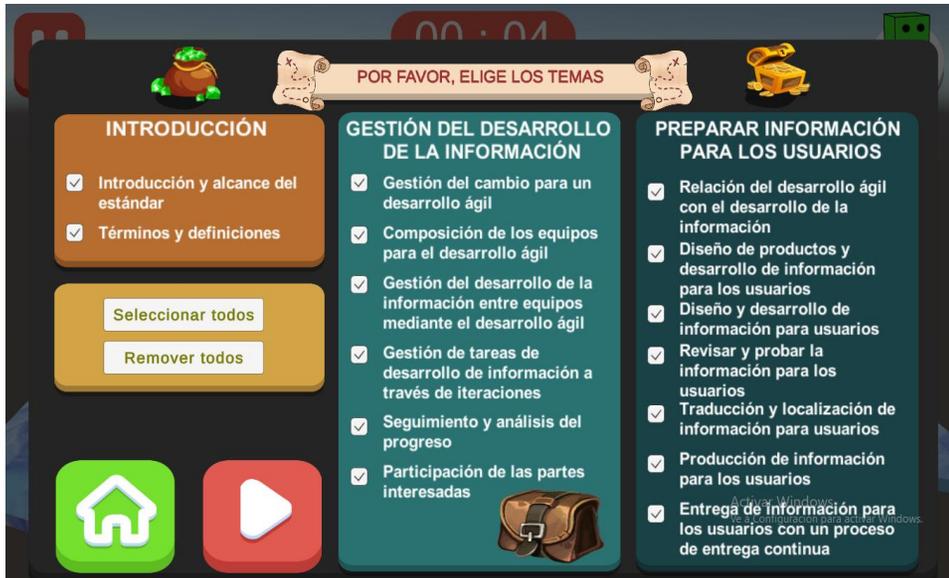
Después de seleccionar la dificultad del juego, el estudiante deberá elegir el color de su *cubitante* (véase Figura 3.12(a)), cuántos jugadores participarán en la competencia y si éstos serán otros estudiantes o si la computadora debe asumir el rol de un jugador más (véase Figura 3.12(b)), y además, elegirá el tipo de dado con el que desea jugar teniendo la posibilidad de escoger un dado de cuatro, seis, ocho o diez caras (véase Figura 3.12(c)).

Una vez que el estudiante termine de configurar sus preferencias en el juego, deberá dar clic en el botón de iniciar para comenzar la partida. Posteriormente, el juego le mostrará un menú con todos los temas principales que se encuentran contenidos en las diferentes partes del estándar ISO/IEC/IEEE 26515:2018, independientemente del nivel que se haya elegido. Es importante mencionar que el profesor puede establecer qué temas deberán incluirse durante la sesión de juego para cubrir la parte práctica de la clase y motivar a los estudiantes a competir. Las preguntas que el guardián del camino formule a los competidores durante la carrera, corresponderán con los temas elegidos. No obstante, el estudiante tiene la opción de volver a la pantalla de inicio (🏠) o comenzar a jugar (▶) (véase Figura 3.13).

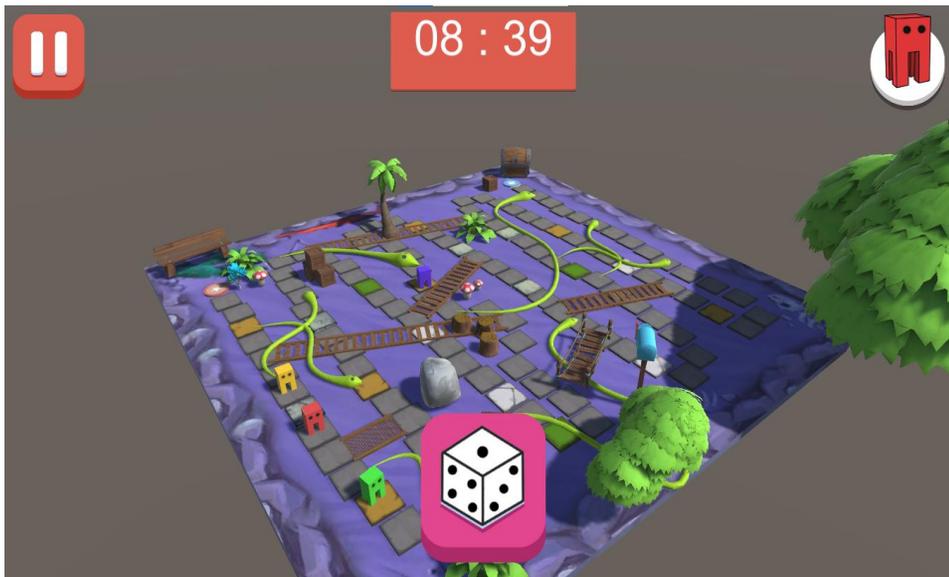


**Figura 3.12.** Pantalla inicial del juego

La competencia inicia mostrando el mapa que los habitantes han creado para sus mejores competidores, quienes deberán dar clic en el icono de dado para lanzarlo y determinar el número de casillas que avanzarán en el camino hacia la meta. En este sentido, el *cubitante* correspondiente avanzará automáticamente en el mapa dependiendo del número que se visualice en la cara superior del dado. Durante toda la competencia, el juego mostrará en la esquina superior derecha al *cubitante* que debe lanzar el dado (*cubitante* rojo en la Figura 3.14).

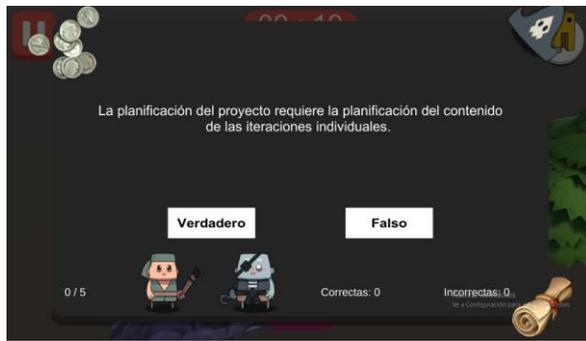


**Figura 3.13.** Selección de temas relacionados con las partes del estándar ISO/IEC/IEEE 26515:2018



**Figura 3.14.** Ejemplo de competencia en desarrollo entre tres jugadores

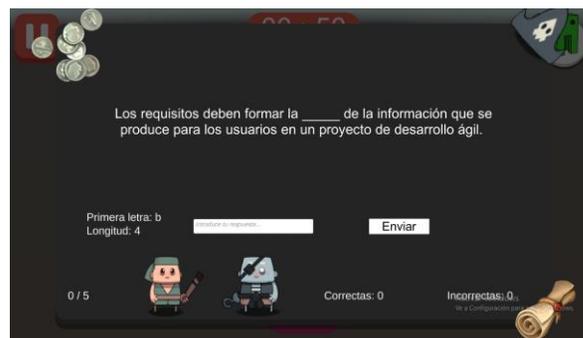
Como se mencionó anteriormente, si un *cubitante* cae fortuitamente en una casilla donde se ubica ya sea una escalera o la cabeza de una serpiente, éste deberá responder preguntas de nivel fácil (a), de nivel medio (b) o de nivel difícil (c) (véase Figura 3.15).



a) Pregunta dicotómica (nivel fácil)



b) Pregunta de opción múltiple (nivel medio)



c) Pregunta abierta (nivel difícil)

**Figura 3.15.** Pantalla inicial del juego

Para llegar a la meta y terminar el juego, el estudiante deberá tirar el dado con el número exacto de casillas restantes, de lo contrario, seguirán tirando hasta que logren dar con el número que necesita para finalizar la carrera. La competencia termina cuando todos los jugadores, a excepción de uno, han llegado a la meta. Cuando esto suceda, el juego muestra una pantalla final con las posiciones en las que llegaron los participantes (véase Figura 3.16). En este punto, el estudiante podrá elegir entre volver al menú principal (🏠) o volver a jugar (🔄).

Finalmente, durante toda la sesión de juego, el estudiante tiene la posibilidad de acceder a un menú de opciones siempre y cuando no se encuentre contestando las preguntas. En dicho menú se podrá pausar, reanudar y reiniciar el juego, además de volver al menú principal y/o quitar el sonido del juego.

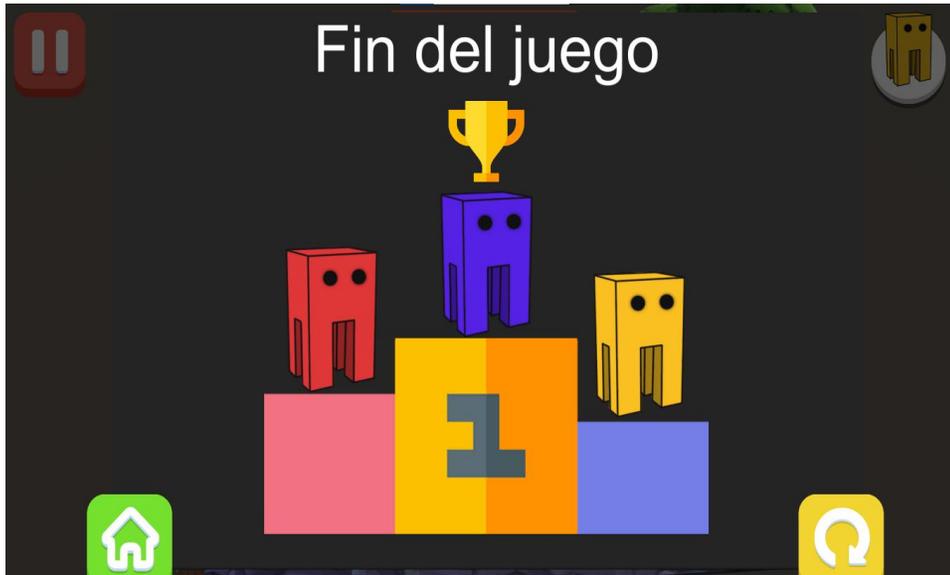
### 3.1.3.2. Personajes

Como ya se explicó anteriormente, los personajes principales del juego son los “*cubitantes*”, o bien, avatares que representan a cada estudiante durante la competencia. El juego permite iniciar una sesión con mínimo un jugador y máximo cuatro. Además, se podrá elegir el color que se desea, teniendo como opciones azul, verde, amarillo y rojo.

### 3.1.4. Fase 1: Diseño de los capítulos

El juego se diseñó para tener un solo capítulo, el cual lleva por nombre `C1_Desarrollo_de_la_informacion` y aborda los retos educativos establecidos en la fase Pre-1. Estos retos son los siguientes:

- Generar una dinámica de aprendizaje basada en el juego que complemente la educación de los estudiantes sobre los métodos ágiles.
- Abordar los procesos y actividades del estándar ISO/IEC/IEEE 26515:2018, con el fin de que los estudiantes comprendan los conceptos fundamentales sobre la documentación en entornos ágiles de desarrollo.
- Relacionar a los estudiantes con las técnicas utilizadas para desarrollar la distinta documentación requerida por el usuario, así como con los diferentes tipos de documentos requeridos en un entorno ágil de desarrollo.
- Fortalecer el aprendizaje de los conceptos teóricos a través de sesiones de juego.



**Figura 3.16.** Ejemplo de competencia en desarrollo entre tres jugadores

Por lo tanto, en este capítulo se desarrolla básicamente toda la historia del juego, involucrando a todos los posibles escenarios que puede haber y todas las iteraciones que el estudiante puede tener con el tablero de juego (i.e., subir o bajar por las casillas, tirar los dados, interactuar con los menús y responder las diferentes preguntas). El capítulo se desarrolla mediante la siguiente secuencia de acciones:

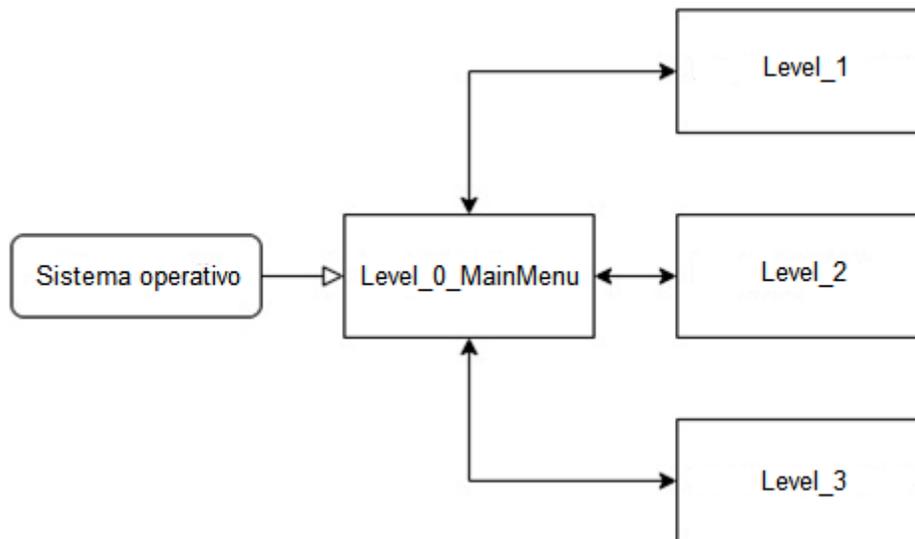
- Una vez que el estudiante haya dado clic en el botón de iniciar, deberá escoger el nivel de dificultad, el número de competidores (i.e., humanos o computadora), el tipo de dado y los temas que desea reforzar con la sesión de juego.
- Cuando se haya terminado de personalizar la sesión, el o los competidores se posicionarán en la casilla de salida del mapa, dependiendo del nivel de dificultad que se haya elegido previamente.
- El o los competidores empezarán a tirar los dados para avanzar en el tablero. Para esto, deberán responder las preguntas cuando sea necesario con el fin de llegar más rápido a la meta.
- El juego termina cuando todos los competidores, a excepción de uno, llegan a la meta.
- Finalmente, se presenta un podio con los tres competidores que hayan concluido satisfactoriamente la carrera y se le da un reconocimiento especial al primer lugar.

### 3.1.5. Fase 2: Diseño de la escena

El capítulo C1\_Desarrollo\_de\_la\_informacion contiene las siguientes cuatro escenas:

- **Level\_0\_MainMenu:** En esta escena se le presentan al estudiante las instrucciones del juego, los niveles que existen, los diversos menús para configurar a los participantes en la carrera y el dado que se desea utilizar.
- **Level\_1:** En esta escena se presenta el mapa para realizar la sesión de juego en su nivel fácil. Dicha escena se diferencia de las dos siguientes básicamente porque, para que el estudiante pueda avanzar o evitar retroceder en el camino, se le formulan preguntas dicotómicas con respuestas de verdadero o falso, facilitándole así el escoger la respuesta correcta.
- **Level\_2:** En esta escena se presenta el mapa para realizar la sesión de juego en su nivel medio. La escena implica que, para que el estudiante avance o evite retroceder en el camino, deberá responder preguntas de opción múltiple, teniendo tres opciones para elegir la respuesta correcta.
- **Level\_3:** En esta escena se presenta el mapa para realizar la sesión de juego en su nivel difícil. Este es el nivel más complicado, puesto que el estudiante deberá introducir con sus propias palabras la respuesta que considere correcta. En este sentido, el estudiante deberá tener cuidado al escribir la palabra que corresponda con la respuesta. Además, el juego muestra, como ayuda para responder la pregunta, la longitud y la primera letra de la palabra respuesta.

De manera general, en la Figura 3.17 se puede visualizar el flujo que existe entre las diversas escenas del juego.



**Figura 3.17.** Flujo de las escenas correspondientes al capítulo diseñado en el juego

A continuación, se describen los escenarios, menús, y objetos con los que el estudiante puede interactuar y la manera en que los retos educativos son abordados en cada una de las escenas descritas con anterioridad.

### 3.1.5.1. Level\_0\_MainMenu

La escena se compone básicamente de tres escenarios:

- El primer escenario corresponde al menú de inicio en el que visualizan el título del juego, las instrucciones para jugar, la configuración del sonido, y la opción de salir o iniciar una nueva partida de juego.
- El segundo escenario corresponde a la acción que se desencadena una vez que el estudiante inicia la sesión de juego. Es decir, se muestran los diferentes niveles de dificultad (i.e., fácil, medio y difícil) en los que se puede jugar para que el estudiante escoja y pruebe sus conocimientos sobre el estándar ISO/IEC/IEEE 26515:2018.
- Finalmente, el tercer escenario muestra un menú para que el estudiante seleccione el número de jugadores que competirán en la partida, si éstos serán jugadores reales o controlados por la computadora, el color de cada jugador y el dado que desea utilizar.

#### 3.1.5.1.1. Diseño del escenario

Los escenarios de esta escena contienen diversos elementos gráficos de interfaz, como son botones y cajas de selección. De manera más específica, se incluyeron los siguientes elementos:

- Botones para abrir los diversos menús (i.e., instrucciones del juego, configuración, cambiar de color al jugador y seleccionar dado).
- Botones para regresar al escenario anterior.
- Botones para avanzar al segundo escenario y, por ende, dar inicio al juego.
- Un botón de salir para cerrar el juego.
- Un botón para poder cerrar los menús.
- Diversas cajas de texto no editables en las que se visualizan las instrucciones del juego, los nombres de los menús a los que el estudiante puede acceder y los niveles de dificultad.
- Tres botones que sirven para seleccionar el nivel de dificultad.
- Un tablero de opciones en el cual el estudiante elige si su competidor será otro humano o la computadora.

#### 3.1.5.1.2. Personajes

Los personajes que se ven involucrados en la escena `Level_0_MainMenu` son aquellos *cubitantes* que el estudiante seleccione para la partida de juego. Como se mencionó anteriormente, se tienen cuatro opciones de avatares de colores: rojo, amarillo, verde y azul. Además, la partida de juego inicia con al menos un competidor y a lo más cuatro.

### 3.1.5.2. Level\_1

En esta escena se compete en el mapa correspondiente al nivel fácil. En este nivel, el estudiante elige los temas del estándar que desea reforzar con la sesión de juego y, considerando su elección, inicia la competencia para llegar a la meta. Es importante recordar que, para avanzar más rápido con una escalera o evitar retroceder en el mapa con una serpiente, el estudiante responderá preguntas dicotómicas con respuestas de verdadero o falso.

### 3.1.5.2.1. Diseño del escenario

El único escenario de esta escena se caracteriza por tener elementos gráficos estáticos y dinámicos con los cuales el estudiante podrá interactuar. En este sentido, dicho escenario contiene los siguientes elementos:

- Un menú que despliega los temas incluidos en el estándar ISO/IEC/IEEE 26515:2018 para que el estudiante seleccione aquellos que desea reforzar con la sesión de juego. Además, se incluyen casillas y botones para personalizar la selección.
- Un botón que el estudiante debe usar para lanzar el dado en el tablero y avanzar automáticamente, como resultado, el número de casillas que corresponda con la cara resultante.
- Un botón de pausa que detiene temporalmente la sesión de juego y, además, muestra en pantalla las diversas opciones del juego.
- Cuatro botones que permiten que el estudiante interactúe con los elementos del menú de pausa (i.e., quitar el sonido, reanudar el juego, reiniciar la sesión del juego, y volver al menú principal).
- Un icono, que figura en la parte superior derecha de la pantalla, para indicar a qué avatar le corresponde el turno de lanzar el dado.
- Un cuadro de texto que muestra el tiempo consumido en la sesión de juego.
- Un menú en el que se muestra la pregunta y las dos opciones de respuesta (i.e., verdadero o falso), además de textos que resumen el número de aciertos, errores y el total de preguntas contestadas.
- Un menú que muestra el podio final al terminar la sesión de juego.
- Diversos elementos que conforman el tablero de juego, entre los que se encuentran las casillas, serpientes, escaleras, arboles, palmeras, arbustos, diversos objetos de madera, champiñones, piedras y los avatares que se hayan elegido para participar en la competencia.

### 3.1.5.2.2. Personajes

Los personajes que actúan en esta escena son el o los avatares que se hayan elegido para realizar la competencia, dependiendo del número de jugadores elegido.

### 3.1.5.2.3. Diálogos y retos del juego

Los retos de esta escena se describen de la siguiente forma:

- Responder las preguntas: El estudiante deberá responder distintas preguntas que se le presentan sobre el estándar ISO/IEC/IEEE 26515:2018. De esta manera, al tiempo que avanza en la competencia irá recordando y asimilando los conocimientos teóricos expuestos en el aula. Por lo tanto, mientras más dure la sesión de juego más se estará en contacto con los conceptos fundamentales del estándar logrando así avanzar de forma más rápida y evitar retroceder en el mapa. Es decir, cuando el estudiante caiga en las casillas para subir por las escaleras o bajar por las serpientes, el juego le muestra un dialogo con la trivía. Así, el estudiante deberá interactuar con los botones para responder las preguntas pudiéndose presentar los siguientes dos casos:
  - El estudiante responde correctamente la pregunta. En este caso, si aún no se han cubierto las tres respuestas correctas, el juego sumará un valor de 1 a las preguntas acertadas y al total de respuestas, para después mostrar la siguiente pregunta. En caso

de que ya se hayan proporcionado las tres respuestas correctas requeridas, el cuadro de diálogo se cerrará y, según sea el caso, el *cubitante* subirá por la escalera o evitará retroceder por la serpiente.

- El estudiante responde erróneamente la pregunta. Para este caso, si no se han proporcionado las tres respuestas correctas, el juego sumará un valor de 1 a las respuestas fallidas, para después mostrar la siguiente pregunta. En caso de que no se hayan proporcionado las tres respuestas correctas, el cuadro de diálogo se cerrará y, según sea el caso, el *cubitante* no subirá por la escalera y avanzará sin usar el atajo o retrocederá en la serpiente y retomará el camino a partir de este punto.

### 3.1.5.3. Level\_2

En esta escena se compete en el mapa correspondiente al nivel medio. En este nivel, el estudiante elige los temas del estándar que desea reforzar con la sesión de juego y, considerando su elección, inicia la competencia. En este caso, el estudiante deberá responder preguntas de opción múltiple con tres respuestas posibles, por lo que deberá elegir solo una de ellas en caso de que caiga en la casilla de escalera o serpiente que harán que avance o retroceda en el camino.

#### 3.1.5.3.1. Diseño del escenario

Al igual que en la escena `Level_1`, la escena `Level_2` se diseñó con un solo escenario que contiene básicamente los mismos elementos:

- Un menú que despliega los temas incluidos en el estándar ISO/IEC/IEEE 26515:2018 para que el estudiante seleccione aquellos que desea reforzar con la sesión de juego. Además, se incluyen casillas y botones para personalizar la selección.
- Un botón que el estudiante debe usar para lanzar el dado en el tablero y avanzar automáticamente, como resultado, el número de casillas que corresponda con la cara resultante.
- Un botón de pausa que detiene temporalmente la sesión de juego y, además, muestra en pantalla las diversas opciones del juego.
- Cuatro botones que permiten que el estudiante interactúe con los elementos del menú de pausa (i.e., quitar el sonido, reanudar el juego, reiniciar la sesión del juego, y volver al menú principal).
- Un icono, que figura en la parte superior derecha de la pantalla, para indicar a qué avatar le corresponde el turno de lanzar el dado.
- Un cuadro de texto que muestra el tiempo consumido en la sesión de juego.
- Un menú en el que se muestra la pregunta y las dos opciones de respuesta (i.e., verdadero o falso), además de textos que resumen el número de aciertos, errores y el total de preguntas contestadas.
- Un menú que muestra el podio final al terminar la sesión de juego.
- Diversos elementos que conforman el tablero de juego, entre los que se encuentran las casillas, serpientes, escaleras, arboles, palmeras, arbustos, diversos objetos de madera, champiñones, piedras y los avatares que se hayan elegido para participar en la competencia. Es importante enfatizar en que, dado que son mapas diferentes, los elementos están distribuidos de diferente manera.

### 3.1.5.3.2. Personajes

De manera similar que en la escena anterior, los personajes que actúan en esta escena son el o los avatares que se hayan elegido para realizar la competencia, dependiendo del número de jugadores elegido.

### 3.1.5.3.3. Diálogos y retos del juego

Los retos siguen la misma estructura de la escena `Level_1`, considerando que ahora se trata de preguntas de opción múltiple. Por lo tanto, el estudiante deberá:

- Responder las preguntas: El estudiante deberá responder acertadamente las distintas preguntas que se le formulen sobre el estándar ISO/IEC/IEEE 26515:2018, por lo que adquirirá un conocimiento más sólido de acuerdo con su avance en las sesiones de juego. Como se mencionó para el caso de la escena anterior, cuando el estudiante caiga en las casillas para subir por las escaleras o bajar por las serpientes, el juego mostrará un diálogo con las preguntas. En este sentido, el estudiante deberá interactuar con los botones de las tres respuestas posibles para responder las preguntas pudiéndose suscitar uno de los siguientes dos casos:
  - El estudiante responde correctamente la pregunta. En este caso, si aún no se han alcanzado las tres respuestas correctas, el juego sumará un valor de 1 por cada respuesta acertada y al total de respuestas, para después mostrar la siguiente pregunta. En caso de que ya se haya cumplido con el requerimiento de proporcionar mínimo tres respuestas correctas, el cuadro de diálogo se cerrará y, según sea el caso, el *cubitante* subirá por la escalera o evitará retroceder por la serpiente.
  - El estudiante responde erróneamente la pregunta. Por otro lado, si aún no se han alcanzado las tres respuestas correctas, el juego sumará un valor de 1 por cada respuesta fallida, para después mostrar la siguiente pregunta. En caso de que no se proporcionen mínimamente tres respuestas correctas, el cuadro de diálogo se cerrará y, según sea el caso, el *cubitante* no subirá por la escalera y avanzará sin usar el atajo o retrocederá en la serpiente y retomará el camino a partir de este punto.

### 3.1.5.4. Level\_3

Por último, en esta escena se compite en el mapa correspondiente al nivel difícil. En este nivel, el estudiante elige también los temas que desea reforzar sobre el estándar y, en base a esta elección, responderá preguntas abiertas para avanzar o evitar retroceder en el mapa. En este sentido, para responder la pregunta mostrada, el estudiante introducirá con sus propias palabras la respuesta que considere correcta. Como una ayuda, el juego le indicará la longitud de la palabra respuesta (i.e., caracteres) y la primera letra de la misma.

#### 3.1.5.4.1. Diseño del escenario

Al igual que las dos escenas anteriores, la escena `Level_3` se diseñó con un solo escenario que contiene los siguientes elementos:

- Un menú que despliega los temas incluidos en el estándar ISO/IEC/IEEE 26515:2018 para que el estudiante seleccione aquellos que desea reforzar con la sesión de juego. Además, se incluyen casillas y botones para personalizar la selección.

- Un botón que el estudiante debe usar para lanzar el dado en el tablero y avanzar automáticamente, como resultado, el número de casillas que corresponda con la cara resultante.
- Un botón de pausa que detiene temporalmente la sesión de juego y, además, muestra en pantalla las diversas opciones del juego.
- Cuatro botones que permiten que el estudiante interactúe con los elementos del menú de pausa (i.e., quitar el sonido, reanudar el juego, reiniciar la sesión del juego, y volver al menú principal).
- Un icono, que figura en la parte superior derecha de la pantalla, para indicar a qué avatar le corresponde el turno de lanzar el dado.
- Un cuadro de texto que muestra el tiempo consumido en la sesión de juego.
- Un menú en el que se muestra la pregunta y las dos opciones de respuesta (i.e., verdadero o falso), además de textos que resumen el número de aciertos, errores y el total de preguntas contestadas.
- Un menú que muestra el podio final al terminar la sesión de juego.
- Diversos elementos que conforman el tablero de juego, entre los que se encuentran las casillas, serpientes, escaleras, arboles, palmeras, arbustos, diversos objetos de madera, champiñones, piedras y los avatares que se hayan elegido para participar en la competencia. Al igual que con el escenario diseñado para la escena `Level_2`, estos elementos están distribuidos de diferente manera puesto que se trata de un mapa diferente.

#### 3.1.5.4.2. Personajes

Los personajes que actúan en esta escena son el o los avatares que se hayan elegido para realizar la competencia, dependiendo del número de jugadores seleccionado.

#### 3.1.5.4.3. Diálogos y retos del juego

Los retos de esta escena se describen de la siguiente forma:

- Responder las preguntas: El estudiante deberá responder acertadamente las distintas preguntas que el juego le presente sobre el estándar ISO/IEC/IEEE 26515:2018, con el objetivo de asimilar los conceptos teóricos expuestos en el aula. Así, cuando el estudiante caiga ya sea en las casillas para subir por las escaleras o bajar por las serpientes, el juego mostrará un diálogo con la trivía. Al responder las preguntas se puede presentar uno de los siguientes dos casos:
  - El estudiante responde correctamente la pregunta. En este caso, si aún no se han proporcionado las tres respuestas correctas, el juego sumará un valor de 1 por cada respuesta acertada y al total de respuestas, para después pasar a la siguiente pregunta. En caso de que ya se haya cumplido con el requerimiento de proporcionar al menos tres respuestas correctas, el cuadro de diálogo se cerrará y, según sea el caso, el *cubitante* subirá por la escalera o evitará retroceder por la serpiente.
  - El estudiante responde erróneamente la pregunta. De lo contrario, si aún no se han alcanzado las tres respuestas correctas, el juego sumará un valor de 1 por cada respuesta fallida para después mostrar la siguiente pregunta. En caso de que no se proporcionen mínimamente tres respuestas correctas, el cuadro de diálogo se cerrará y, según sea el caso, el *cubitante* no subirá por la escalera y avanzará sin usar el atajo o retrocederá en la serpiente y retomará el camino a partir de este punto.

### 3.1.6. Fase 3: Identificación de los retos educativos y su evaluación

A partir de la perspectiva planteada en las secciones anteriores, se propusieron cuatro retos educativos que el juego debía abordar para que estudiantes de nivel licenciatura pudieran relacionarse con el estándar ISO/IEC/IEEE 26515:2018 de manera entretenida. Por lo tanto, fue necesario considerar mecánicas y dinámicas de juego que permitieran que los estudiantes incrementarán su motivación y compromiso para relacionarse con los conceptos definidos por el estándar. De acuerdo con García et al. (2020), las mecánicas de juego representan componentes funcionales que definen diferentes acciones, comportamientos, y mecanismos de control que promueven la motivación de los estudiantes, mientras que las dinámicas de juego determinan sus reacciones como respuesta al uso de las mecánicas implementadas y, de hecho, con estas reacciones se trata de satisfacer sus necesidades y deseos fundamentales durante las sesiones de juego.

En este sentido, en cada escena del juego, el estudiante debe interactuar con el mapa a través de las mecánicas implementadas y recibe realimentación, como consecuencia de sus acciones, mediante las dinámicas introducidas. Teniendo esto en consideración, los retos educativos que el estudiante debe superar están intrínsecamente relacionados con las mecánicas y dinámicas implementadas en el juego de “Serpientes y escaleras”. Con el objetivo de detallar la manera en que tales mecánicas y dinámicas de juego fueran consideradas e implementadas en “*Serpientes y escaleras: Un juego de soporte al aprendizaje del estándar ISO/IEC/IEEE 26515:2018*”, a continuación se proporciona una explicación más específica de cada decisión tomada al respecto:

- **Concurso (mecánica):** Cuando se trata de competir, los jugadores buscan obtener las puntuaciones más altas para hacerse acreedores a premios y/o reconocimientos. Así, en el contexto del juego propuesto, el concurso es la mecánica fundamental puesto que es la que se vincula directamente con su tipo (trivia), dado que el estudiante debe responder preguntas sobre el estándar ISO/IEC/IEEE 26515:2018, el reto educativo de mayor importancia. En este sentido, el juego promueve el concurso entre los participantes puesto que deberán responder correctamente las preguntas para avanzar de forma más rápida por el mapa y llegar a la meta. Por lo tanto, a través de esta mecánica, el estudiante refuerza su conocimiento sobre el estándar y, además, visualiza claramente cuáles son los temas que debe repasar para entender correctamente cómo se desarrolla la información para los usuarios en un entorno ágil.
- **Niveles (mecánica):** Los niveles suelen presentarse al jugador mediante rangos de puntos, complejidades, o incluso retos. En el contexto del juego presentado en esta tesis, la trivia se implementó con tres diferentes niveles de dificultad que sirven de apoyo para que el estudiante pueda reforzar el conocimiento sobre el estándar ISO/IEC/IEEE 26515:2018 de formas distintas. Los niveles implican cierto incremento de dificultad en las preguntas y la forma en que se debe dar una respuesta. Aunado a lo anterior, el estudiante se ve involucrado en diferentes escenarios de juego, haciendo así su aprendizaje más entretenido.
- **Realimentación (mecánica):** Al jugar, el jugador espera a que el juego reaccione a sus acciones con el fin de motivar su desarrollo personal. En este contexto, el juego propuesto proporciona realimentación tanto positiva como negativa a los estudiantes. Es decir, si el estudiante cae en la casilla donde se ubica una escalera y responde correctamente al menos tres de cinco preguntas que el juego le hará, considerando los temas elegidos al inicio de la sesión de juego, entonces se generará una realimentación positiva visual que mostrará al *cubitante* tomando el atajo a través de la escalera. En caso contrario, el estudiante recibirá un mensaje por cada respuesta incorrecta y deberá avanzar en el tablero sin el uso de atajos. De manera similar, el estudiante recibe realimentación negativa visual si, en caso de caer en una casilla donde se ubica una serpiente, responde incorrectamente a las preguntas, puesto que se mostrará cómo

el *cubitante* retrocede a la cola de la serpiente, perdiendo así la oportunidad de aventajar a sus contrincantes.

- **Competencia (dinámica):** Durante la sesión de juego, todos los participantes aspiran a alcanzar un mismo objetivo con el fin de obtener el reconocimiento final de superioridad. Por lo tanto, el juego propuesto implementó un enfrentamiento sano entre los estudiantes con el fin de determinar quién de ellos ha entendido mejor los conceptos fundamentales del estándar ISO/IEC/IEEE 26515:2018 y motivarlos continuamente a mejorar. De esta manera, el *ranking* o la clasificación en el uso de un juego serio es lo que lo hace competitivo, es decir, para que exista la competitividad el juego premia a quién haya llegado más rápido a la meta dándole al ganador la sensación de que fue mejor que sus contrincantes en la partida.
- **Estatus (dinámica):** A cualquier jugador le gusta ver su evolución al jugar, ya sea a través de los puntos ganados, los niveles que pueda superar o por el mero reconocimiento. Esto hará que la autoestima se eleve y se genere motivación por seguir jugando. Así, el juego propuesto proporciona un entorno libre de riesgos para que los estudiantes recorran una isla, a través de sus avatares, con el fin de alcanzar más rápido la meta y obtener el reconocimiento de los habitantes de Cubilandia.
- **Logro (dinámica):** Puede parecer similar a la recompensa; sin embargo, la recompensa permite que el jugador obtenga algo a cambio por su desempeño durante la sesión de juego, mientras que un logro es más una satisfacción personal por haber conseguido un objetivo o ser reconocido socialmente. Esta dinámica está presente en el juego propuesto puesto que mientras más competencias gane un estudiante, se logrará ser el *cubitante* más veloz de la isla.
- **Recompensa (dinámica):** Todo juego tiene la finalidad de hacer sentir al jugador que, en caso de mostrar un buen desempeño, recibirá recompensas tangibles o virtuales en forma de reconocimiento. En este sentido, el juego propuesto incluye al final de cada competencia un podio cuyo objetivo es recompensar al estudiante y motivarlo a esforzarse más en las futuras sesiones de juego. De esta manera, se estableció un sistema de competitividad saludable para los estudiantes, quienes al jugar de manera constante pueden reforzar sus conocimientos teóricos sobre el estándar ISO/IEC/IEEE 26515:2018.

La Tabla 10 resume la información relacionada con los retos educativos y las mecánicas y dinámicas de juego implementadas.

**Tabla 10.** Retos educativos, mecánicas y dinámicas de juego

<b>Reto educativo</b>	<b>Forma de evaluación</b>	<b>Mecánica</b>	<b>Dinámica</b>
1. Mejorar la comprensión de los estudiantes sobre los conceptos fundamentales del estándar ISO/IEC/IEEE 26515:2018.	Dado que es un juego de trivia, el estudiante deberá responder a tres preguntas, al menos, de cinco que le formula el juego en caso de caer en casillas de escaleras o serpientes.	Realimentación	Recompensa
2. Establecer una dinámica de aprendizaje a través de la cual los estudiantes sean testigos de su correcto entendimiento de los conceptos fundamentales del estándar.	En caso de responder correctamente a las preguntas realizadas, el estudiante adelantará a sus competidores en la escalera para alcanzar más rápido la meta o bien no retrocederá por la serpiente y evitará, como consecuencia, perder la ventaja ganada.	Concurso, realimentación, movimiento de celdas, tablero modular, tirar dados	Logro
3. Establecer una dinámica de aprendizaje a través de la cual los	En caso de no responder correctamente a las preguntas	Concurso, realimentación,	Logro

<b>Reto educativo</b>	<b>Forma de evaluación</b>	<b>Mecánica</b>	<b>Dinámica</b>
estudiantes identifiquen que no está entendiendo correctamente los conceptos fundamentales del estándar.	realizadas, el estudiante no podrá subir por la escalera para adelantar a sus competidores y posiblemente llegue más tarde a la meta o bien retrocederá en la serpiente y, como consecuencia, perderá la ventaja ganada.	movimiento de celdas, tablero modular, tirar dados	
4. Motivar a los estudiantes para alcanzar un aprendizaje más activo.	El estudiante que llegue más rápido a la meta será el ganador de la competencia y subirá al podio del primer lugar.	Niveles	Competencia, estatus, recompensa

Considerando lo anteriormente expuesto, la evaluación del juego aborda los retos educativos definidos en la Tabla 10. De esta forma, a través de las sesiones de juego, se evalúa si los estudiantes logran o no comprender de forma satisfactoria los métodos y técnicas utilizadas para el desarrollo de la información en entornos ágiles. Es decir, mientras el estudiante esté en una sesión de juego y responda las preguntas de la trivía, podrá percatarse de cuáles son los temas del estándar que ya domina y cuáles son los que necesita reforzar posteriormente. Debido a que el juego no establece un tiempo límite para que los estudiantes respondan cada pregunta, éstos pueden consultar al profesor para identificar la respuesta correcta, fomentando así la realimentación constante en tiempo de ejecución. De una manera más personal, el estudiante podrá anotar la pregunta que ha fallado para después investigar por su cuenta propia la respuesta, si así lo desea, directamente en el estándar. Así, se busca aumentar la motivación de los estudiantes para mejorar su comprensión sobre los estándares del proceso de software y erradicar la percepción de que éstos son documentos demasiado técnicos e imposibles de entender.

### **3.1.7. Fase 4: Identificación/clasificación de emociones**

Todas las emociones que emanen durante las sesiones del juego son importantes para la correcta evaluación del impacto que éste ha tenido en los estudiantes. Teniendo esto en consideración, Goleman (1996) argumentó que las emociones se podían definir como respuestas psicofisiológicas a estímulos internos o externos que pueden variar en intensidad y duración. En este sentido, considerando las reflexiones y los aprendizajes que los estudiantes pueden experimentar como consecuencia de la experiencia con el juego, en las diferentes escenas que lo conforman se generan diversas emociones que se describen a continuación.

#### **3.1.7.1. Emociones presentes en la escena Level\_0\_MainMenu**

En esta escena se pretendió generar las siguientes emociones en el estudiante:

- **Interés:** Es una de las emociones clave en el desarrollo del juego, puesto que al mostrar al estudiante elementos gráficos llamativos y permitirle la personalización de algunos de éstos, se intentó crear un entorno atractivo que permitiera mantener los niveles adecuados de compromiso para realizar sesiones de juego de principio a fin y se logre, como consecuencia, completar con éxito los retos y objetivos planteados en el juego.
- **Intriga:** Al presentarle al estudiante únicamente el título y las instrucciones del juego al inicio de la sesión, se espera que éste experimente intriga por saber qué deberá hacer durante el juego.
- **Indiferencia:** No se descarta que algunos estudiantes no sientan interés por el juego creado y que, por ende, muestren una actitud negativa al permanecer en la sesión por simple compromiso.

### 3.1.7.2. Emociones presentes en las escenas Level\_1, Level\_2 y Level\_3

Para estas escenas del juego se consideró el propiciar las siguientes emociones en el estudiante:

- **Motivación:** Los juegos pueden estimular la motivación intrínseca de los estudiantes, ya que algunos de éstos a menudo se obligan a mostrar un buen comportamiento en las sesiones de juego impulsados por un deseo interno de no fracasar. Cuando se presenta, esta motivación puede generar que el estudiante desee participar activamente en el juego para alcanzar sus objetivos. En este sentido, el juego propuesto en esta tesis se diseñó para mantener o incrementar esta motivación de las siguientes formas:
  - **Realimentación inmediata:** El estudiante recibe, de manera inmediata, realimentación visual positiva y/o negativa de acuerdo con su desempeño al responder las preguntas. De esta manera, se pretendió motivarlo a mejorar cuando falle en una respuesta o recompensarlo cuando acierte.
  - **Objetivos claros:** El juego establece objetivos y mecánicas simples con el fin de que el estudiante entienda claramente la meta a alcanzar. De esta manera, a través del reto de responder correctamente las preguntas formuladas por el juego, el estudiante se motiva constantemente por adelantar a sus compañeros mediante el uso de las escaleras.
  - **Recompensas y logros:** El juego ofrece recompensas al estudiante cuando responde correctamente las preguntas. Así, estos incentivos pueden motivarlo a seguir adelante y esforzarse en superarse en cada sesión de juego.
- **Diversión:** Uno de los objetivos principales que se buscó al diseñar el juego es que el estudiante se divirtiera al mismo tiempo que reafirmaba sus conocimientos teóricos sobre el estándar y que, además, deseara seguir jugando hasta terminar la competencia o incluso que quisiera jugar de nuevo sin necesidad de sentirse forzado.
- **Competitividad:** El juego permite hasta cuatro jugadores por sesión de juego, por lo que se espera que los estudiantes puedan sentir la emoción de competir contra sus compañeros y el deseo de mejorar su propio rendimiento.
- **Curiosidad:** Al tener una base de datos de preguntas, se pretendió que el estudiante sintiera curiosidad por jugar varias veces en los diferentes niveles incluidos en el juego, reforzando así el conocimiento adquirido en el aula.
- **Sorpresa:** Cuando un usuario responda correcta o incorrectamente una pregunta, es posible que experimente sorpresa por el movimiento que realiza el *cubitante* en el tablero.
- **Felicidad:** Al obtener una realimentación constante, el estudiante se puede sentir feliz cuando logre avanzar en el tablero más rápido que sus contrincantes. De ser así, se estaría generando además una sensación de éxito y progreso.
- **Colaboración:** A pesar de que no fue diseñado para jugarse colaborativamente, dado que el juego permite múltiples competidores, se puede dar el caso en que los mismos estudiantes colaboren para responder una pregunta.
- **Decepción:** Se puede dar el caso de que el juego propuesto no satisfaga las expectativas de los estudiantes y profesores, por lo que ambos se podrían sentir obligados a terminar la sesión.
- **Enojo:** Se puede también dar el caso de que un estudiante se sienta frustrado por no responder correctamente las preguntas y quedarse rezagado con frecuencia durante la competencia.
- **Impotencia:** Finalmente, puede ocurrir que un estudiante no acierte en las respuestas y sienta frustración al no poder avanzar en las escaleras o retroceder en las serpientes, desmotivándolo así para continuar motivado durante el resto de la competencia.

### 3.1.8. Fase 5: Diseño de la adaptación

El estudiante tiene las siguientes opciones de personalización del juego:

- Color de avatar: El estudiante podrá elegir, entre amarillo, rojo, azul y verde, el color de su avatar.
- Dado: El estudiante podrá elegir el dado con el que desea jugar, teniendo como opciones dados de cuatro, seis, ocho y diez caras.
- Tipo y número de jugadores: De manera adicional, se tiene la opción de poder jugar con una persona real o con la propia máquina simulando un jugador más, permitiendo conducir una sesión de juego de hasta cuatro participantes.
- Sonido: El juego permite jugar con o sin sonido.
- Dificultad: El juego contiene tres niveles de dificultad: fácil, medio y difícil. Cada nivel involucra diferentes tipos de preguntas:
  - Fácil: En este nivel, se presentan preguntas dicotómicas con respuestas de verdadero o falso.
  - Medio: En el nivel medio se presentan preguntas de opción múltiple, teniendo tres opciones distintas de respuestas.
  - Difícil: Para el nivel más difícil el estudiante deberá introducir su propia respuesta haciendo uso del teclado.

### 3.1.9. Fase 6: Diseño de la colaboración

El diseño del juego permite la incorporación de actividades en equipo o que requieran colaboración, permitiendo así que los profesores organicen sesiones de juego de uno a cuatro jugadores (un representante por cada equipo). Aunado a eso, también se pueden realizar sesiones de un estudiante contra la propia computadora, por ejemplo, tres jugadores simulados por la computadora.

#### 3.1.9.1. Implementación del juego

El juego implementado en esta tesis se cataloga en la categoría de juegos 3D, los cuales se caracterizan por crear entornos, personajes y objetos con profundidad y realismo en tres dimensiones: el eje  $x$ ,  $y$ , y  $z$ . Una característica de este tipo de juegos es que ofrecen una experiencia más inmersiva y visualmente atractiva, ya que simulan el mundo tridimensional de una manera que se asemeja más a la realidad. El desarrollo del juego siguió las fases descritas a lo largo de este capítulo de la tesis, por lo que fue necesario definir las herramientas y el lenguaje de programación a utilizar como fase principal, los cuales se presentan a continuación.

#### 3.1.9.2. Herramientas y lenguaje de programación

Actualmente se encuentran disponibles diversos motores de videojuegos diseñados con el propósito de simplificar la integración de elementos como contenido, diseño y representación en el desarrollo de videojuegos, agilizando así todo el proceso. En este contexto, el juego llamado “*Serpientes y escaleras: Un juego de soporte al aprendizaje del estándar ISO/IEC/IEEE 26515:2018*”

fue desarrollado en la plataforma de Unity 3D®<sup>5</sup> debido a sus capacidades para facilitar la incorporación de archivos tridimensionales y su manipulación en el mismo entorno de desarrollo. Además de esto, Unity 3D® permite infundir dinamismo en los objetos del juego, como personajes, casillas, arboles, entre otros, mediante la implementación de *scripts* en lenguaje C# o JavaScript, ya que ambas son compatibles con esta herramienta. En esta tesis se decidió utilizar C# como lenguaje principal de programación.

### 3.1.9.3. Desarrollo

Como se mostró a lo largo de este capítulo de la tesis, el proceso de desarrollo inició con la identificación de los desafíos educativos que el juego debe abordar, seguido por la creación de la narrativa específica para el mismo. Con base en esta información, se procedió a la planificación detallada de los capítulos y las escenas que conformarían la estructura del juego. En este contexto, la escena de apertura del juego presenta un menú de opciones que permiten al usuario iniciar una nueva partida, leer las instrucciones, configurar el sonido y salir del juego (véase Figura 3.10).

Una vez que el estudiante inicia el juego, tiene la oportunidad de personalizar distintos elementos, como son el color del jugador, el número y tipo de jugadores que participarán en la competencia, el dado con el que se desea jugar, el nivel de dificultad para la competencia, o incluso iniciar la partida, configurar el sonido, o salir del juego (véase Figuras 3.11 y 3.12). Después de que se terminó de personalizar la sesión de juego, el estudiante será ubicado en la escena correspondiente al nivel de dificultad que escogió, para posteriormente elegir los temas del estándar que desea reforzar con la sesión en curso (véase Figura 3.13). Una vez escogidos, todos los avatares de los competidores son ubicados en la casilla de salida para dar inicio formal a la competencia. El(la) o lo(a)s estudiantes deberán tirar los dados para avanzar por el tablero hacia la línea de meta, encontrándose por el camino escaleras o serpientes que servirán de atajos o trampas, respectivamente. En este sentido, el(la) estudiante también dispone de un cronómetro, que se muestra en la parte superior central de la pantalla, con el tiempo que ha durado la sesión de juego en curso y una imagen del avatar con el turno para tirar el dado (véase Figura 3.14). Estas casillas clave mostrarán diálogos de texto con los que los estudiantes deberán interactuar para responder una serie de preguntas (véase Figura 3.15). Estos diálogos se implementaron considerando la siguiente clasificación:

- Nivel fácil: Preguntas dicotómicas con respuestas de verdadero o falso.
- Nivel medio: Preguntas de opción múltiple.
- Nivel difícil: Preguntas abiertas.

Finalmente, se implementó un sistema de podio en el juego para mostrar al ganador de la competencia una vez que todos los competidores, a excepción de uno, hayan alcanzado la línea de meta (véase Figura 3.16).

## 3.2. Comentarios finales sobre el desarrollo del juego serio

Tal como se explicó a lo largo de este capítulo, considerando el antiguo juego de “Serpientes y escaleras”, se diseñó e implementó un juego serio llamado “*Serpientes y escaleras: Un juego de*

---

<sup>5</sup> Unity es un motor de videojuego multiplataforma creado por *Unity Technologies* que, además, está disponible como plataforma de desarrollo de juegos para Microsoft Windows, Mac OS, y Linux. La plataforma de desarrollo tiene soporte de compilación con diferentes tipos de plataformas. Para más información, por favor consulte: <https://unity.com/es>.

*soporte al aprendizaje del estándar ISO/IEC/IEEE 26515:2018*” como una herramienta de apoyo para aquellos profesores que imparten materias relacionadas con el desarrollo de software mediante el uso de los métodos ágiles y que, por ende, deben abordar temas sobre el desarrollo de la información para los usuarios en tales entornos de desarrollo. En este contexto, el profesor deberá seguir empleando el material respectivo para transmitir el conocimiento a los estudiantes de manera teórica dentro de las aulas de clases siguiendo su enfoque educativo tradicional. Es decir, el juego presentado en este capítulo de la tesis pretende complementar la enseñanza de una forma más dinámica y divertida.

Es importante mencionar que, bajo el enfoque que sigue el juego serio, los estudiantes no necesariamente deben estar bajo la supervisión directa de un profesor, ya que este recurso educativo se ha diseñado de manera que los estudiantes puedan aprender de forma autónoma construyendo ellos mismos su conocimiento. No obstante, como se mencionó anteriormente, si el profesor desea organizar competencias entre equipos de estudiantes es recomendable que siempre esté presente para establecer y vigilar las reglas de juego.

En el siguiente capítulo de esta tesis se presenta una evaluación empírica que se realizó con el objetivo de analizar la motivación, experiencia de uso y aprendizaje de los estudiantes participantes al reforzar su conocimiento sobre los elementos fundamentales del estándar ISO/IEC/IEEE 26515:2018 - *Systems and software engineering —Developing information for users in an agile environment* mediante la realización de sesiones con el juego propuesto. De manera similar, se evaluó tanto la percepción de los profesores participantes sobre el juego como recurso educativo, como su efectividad para mejorar el desempeño académico de los estudiantes.



## 4. Resultados de la evaluación empírica

En este cuarto capítulo se presentan tanto el diseño como la realización de un caso de estudio que incluyó a estudiantes de nivel licenciatura en un contexto educativo real, y cuyo fin fue el medir la eficacia del juego desarrollado. Aunado a lo anterior, se presenta un análisis de los resultados obtenidos.

### 4.1. Participantes

Un estudiante egresado del programa de la Licenciatura de Ingeniería en Computación de la UTM, que es el tesista, diseñó y desarrolló el juego serio llamado “*Serpientes y escaleras: Un juego de soporte al aprendizaje del estándar ISO/IEC/IEEE 26515:2018*” que fue puesto bajo evaluación. Además, este mismo estudiante diseñó los cuestionarios de evaluación que fueron aplicados y participó en el análisis de la información recogida junto con el director y codirectora de esta tesis.

Un total de 20 estudiantes del noveno semestre del programa de la Licenciatura de Ingeniería en Computación de la UTM fueron parte de la evaluación empírica del juego serio. De este conjunto, 18 estudiantes estaban cursando la especialidad de “Ingeniería de Software”, mientras que dos se encontraban cursando las materias optativas de la especialidad en “Inteligencia Artificial”. No obstante, el total de los estudiantes había cursado previamente las materias de “Ingeniería de Software” y “Desarrollo Ágil de Software”. Estos estudiantes fueron separados en dos grupos: el grupo de control y el grupo experimental. Cada grupo se conformó con diez estudiantes que fueron seleccionados de manera aleatoria.

Aunado a esto, dos profesores-investigadores de la División de Estudios de Posgrado de la UTM participaron en el caso de estudio impartiendo las clases teóricas a todos los estudiantes de ambos grupos. Durante las clases, que se realizaron en las aulas del Laboratorio de Metodologías de Desarrollo e Inteligencia Artificial de la División de Estudios de Posgrado de la UTM durante cuatro semanas, se expusieron temas importantes sobre el desarrollo ágil de software y el estándar ISO/IEC/IEEE 26515:2018. De igual forma, estos dos participantes aplicaron las evaluaciones pertinentes con el fin de determinar si el juego serio logró cumplir con los objetivos propuestos en esta tesis.

Finalmente, cinco profesores que imparten materias relacionadas con la Ingeniería de Software en tres universidades (NovaUniversitas, Universidad de la Sierra Juárez y Universidad del Istmo) incorporadas al Sistema de Universidades Estatales de Oaxaca (SUNEO) respondieron un cuestionario para determinar su percepción sobre el juego propuesto después de haberlo jugado. Estos profesores, quienes tienen una sólida experiencia en la enseñanza de la Ingeniería de Software a nivel universitario, recibieron el juego para participar en sesiones de competencia contra la computadora.

## 4.2. Instrumentos

Se desarrollaron tres instrumentos de evaluación. En primera instancia, se diseñó un cuestionario dirigido a ambos grupos de estudiantes que se conformó de 15 preguntas abiertas para determinar el conocimiento sobre temas específicos recomendados para la enseñanza de los métodos ágiles y los procesos de desarrollo de la documentación en entornos ágiles. Es importante mencionar que estos temas fueron derivados de las directrices curriculares establecidas en los planes de estudio tanto del programa de Licenciatura en Informática como de la Ingeniería en Computación de las universidades pertenecientes al SUNEО. Además, otras 15 preguntas abiertas fueron agregadas al cuestionario para determinar el nivel de conocimiento de los estudiantes sobre el estándar ISO/IEC/IEEE 26515:2018, después de haber realizado las clases teóricas y, en el caso del grupo experimental, las sesiones de juego. El cuestionario resultante se muestra en la Tabla 11.

**Tabla 11.** Cuestionario para determinar el nivel de comprensión de los estudiantes

Pregunta	Respuesta
1. ¿Qué conocimiento deben de tener los ingenieros de software para llevar a cabo de manera correcta los procesos definidos por los métodos ágiles?	
2. ¿Cuál es la primera medida de éxito en los métodos ágiles?	
3. ¿Quiénes son los integrantes de un equipo de desarrollo ágil?	
4. Menciona al menos dos métodos ágiles populares.	
5. ¿Qué es un “ <i>Product Owner</i> ” en Scrum y cuál es su función principal?	
6. ¿Cómo se lleva a cabo la planificación de <i>sprints</i> en Scrum?	
7. ¿A qué se refiere el concepto de “entrega continua” en los métodos ágiles?	
8. ¿Cuál es la diferencia clave entre Scrum y Kanban?	
9. ¿Cuál es la principal diferencia que existe entre la documentación tradicional y la documentación ágil?	
10. ¿Qué es una “historia de usuario” en un contexto ágil y cómo se documenta?	
11. ¿Cuál es el significado de la definición de “hecho” en un contexto ágil y cómo se documenta?	
12. ¿En qué proceso del desarrollo ágil se obtiene la documentación del software?	
13. ¿Cómo se documentan los resultados de las retrospectivas en un proyecto ágil?	
14. ¿Cómo se garantiza la trazabilidad de los requisitos en un proyecto ágil?	
15. ¿Cuál es la diferencia entre la documentación que se genera para el equipo de desarrollo y la documentación que es para los <i>stakeholders</i> ?	
16. ¿Qué información valiosa proporcionan los casos de uso a los desarrolladores de la información?	
17. ¿Cuál es el propósito del estándar ISO/IEC/IEEE 26515:2018 - <i>Systems and software engineering — Developing information for users in an agile environment</i> ?	
18. ¿Cuáles son los tipos de usuarios a los que se dirige el estándar ISO/IEC/IEEE 26515:2018?	
19. ¿Cuántas partes conforman al estándar ISO/IEC/IEEE 26515:2018?	

Pregunta	Respuesta
20. ¿En qué momento se pueden realizar las pruebas de usabilidad sobre la información que se entregará a los usuarios?	
21. ¿Qué beneficios se obtienen de desarrollar la información en paralelo con el software?	
22. ¿Cuál es el propósito principal de incluir información complementaria en un plan para el desarrollo de la información?	
23. Dado que la comunicación en el desarrollo ágil es en tiempo real, ¿qué deben hacer los desarrolladores de la información para estar informados?	
24. ¿Por qué es importante que en todos los proyectos se revise y <i>testee</i> la información que se entregará a los usuarios?	
25. ¿Quiénes son los stakeholders que están directamente involucrados en un plan de desarrollo de la información?	
26. ¿Quién debe enumerar y dimensionar las tareas relacionadas con el desarrollo de la información antes del inicio de cada iteración?	
27. ¿Qué técnica menciona el estándar ISO/IEC/IEEE 26515:2018 como una alternativa para desarrollar la información para el usuario cuando no se cuenta con las especificaciones del diseño?	
28. ¿Qué sugiere el estándar ISO/IEC/IEEE 26515:2018 sobre la información para los usuarios que es producida en iteraciones pasadas?	
29. ¿Cuál es la longitud y el nivel de detalle de un escenario?	
30. ¿A qué fase del ciclo de vida se limita el estándar ISO/IEC/IEEE 26515 para guiar al usuario en la creación de la información para los usuarios?	

Por otro lado, se modificó el cuestionario de medición del MEEGA+ (Petri et al., 2019) para medir el rendimiento de los juegos serios, tanto digitales como no digitales, teniendo en cuenta la evaluación de ocho dimensiones: *usabilidad* (i.e., efectividad, eficiencia y satisfacción con que un estudiante logra objetivos específicos mientras juega), *confianza* (i.e., capacidad de lo(a)s estudiantes para progresar en el aprendizaje de contenidos educativos a través de su esfuerzo y habilidad), *desafío* (i.e., qué tan desafiante es el juego con relación al nivel de competencia del(a) estudiante), *satisfacción* (i.e., sentimiento de los estudiantes cuando perciben que el esfuerzo dedicado conduce al aprendizaje), *diversión* (i.e., sentimientos de placer, felicidad, relajación y distracción de lo(a)s estudiantes), *atención enfocada* (i.e., atención, concentración enfocada, absorción y disociación temporal de lo(a)s estudiantes mientras juegan), *relevancia* (i.e., la percepción de los estudiantes sobre si la propuesta educativa es consistente con sus objetivos o no y si es posible vincular el contenido a su futuro profesional o académico) y *aprendizaje percibido* (i.e., percepciones del efecto general del juego en el aprendizaje de lo(a)s estudiantes en la materia). La medición se realiza utilizando una escala tipo Likert de cinco puntos para las alternativas de respuestas, que son “Totalmente en desacuerdo” (TD) = -2.0, “En desacuerdo” (ED) = -1.0, “Neutral” (N) = 0.0, “De acuerdo” (DA) = 1.0 y “Totalmente de acuerdo” (TA) = 2.0. Esto implica que las respuestas subjetivas se convirtieron en valores numéricos que facilitan un análisis descriptivo a través del cálculo de la media (M) y la desviación estándar (SD). El instrumento de evaluación se muestra en la Tabla 12.

**Tabla 12.** Instrumento de medición de MEEGA+ (traducido y adaptado de Petri et al., (2019))

Dimensión		Afirmaciones	TD	ED	N	DA	TA	M	S
Usabilidad	Estética	1. El diseño del juego es atractivo con relación a su interfaz, gráficos, etc.							
		2. Las fuentes de texto y los colores usados en el juego me parecen adecuados y consistentes.							
		3. Considero que no se requiere de un tutorial para comprender correctamente la temática del juego.							
	Capacidad de aprendizaje	4. La jugabilidad se puede interpretar fácilmente.							
		5. La mayoría de las personas podrían aprender rápidamente los comandos del juego.							
	Operabilidad	6. El juego puede jugarse fácilmente.							
		7. Las reglas del juego son claras y concisas.							
	Accesibilidad	8. El tamaño y estilo de las fuentes usadas en el juego facilitan la lectura de los textos.							
		9. Los colores usados son los adecuados para la temática del juego.							
		10. La apariencia del juego (fuentes, colores, etc.) se puede personalizar de acuerdo con las preferencias de un(a) jugador(a).							
	Protección de errores	11. El juego impide que el(la) jugador(a) cometa errores.							
		12. Es fácil recuperarse de errores cometidos en el juego.							
Confianza	13. Desde el inicio del juego da la impresión de que es fácil de jugar.								
	14. Tanto el contenido como la estructura del juego dan al(a) jugador(a) la confianza de que aprenderá algo nuevo.								
Desafío	15. La dificultad del juego es la apropiada.								
	16. El ritmo de las misiones (situaciones, variaciones, nuevos obstáculos, etc.) varía adecuadamente en el juego conforme el tiempo avanza.								
	17. 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 ofrece una sensación de logro.								
	19. Con esfuerzo es posible sortear los obstáculos del juego e ir avanzando.								
	20. El juego genera satisfacción con el aprendizaje obtenido.								
	21. Recomendaría este juego a mis amigo(a)s o compañero(a)s.								
Diversión	22. Las personas pueden divertirse con el juego.								

Dimensión	Afirmaciones	TD	ED	N	DA	TA	M	S
	23. Algunos elementos del juego (diálogos, misiones, imágenes, etc.) pueden hacer que lo(a)s jugadore(a)s se rían.							
Atención enfocada	24. Hay aspectos del juego que pueden llamar la atención del(a) jugador(a) desde el inicio.							
	25. La concentración en las misiones del juego hace que se pierda la noción del tiempo.							
	26. Es posible que un(a) jugador(a) se olvide de su entorno mientras juega.							
Relevancia	27. Los contenidos mostrados en el juego son de relevancia para la clase teórica.							
	28. Existe una relación clara entre el juego y los objetivos de aprendizaje.							
	29. El juego es una forma adecuada de fortalecer el aprendizaje sobre el tema.							
	30. Es posible que los estudiantes prefieran reforzar sus conocimientos con este juego que con otras formas tradicionales (libros, diapositivas, etc.).							
Aprendizaje percibido	31. El juego puede contribuir a mejorar el aprendizaje de los tópicos vistos en la clase teórica.							
	32. La enseñanza brindada por el juego fue más eficiente para el aprendizaje de los tópicos vistos 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.

Finalmente, con el objetivo de aumentar la validez de la evaluación empírica sobre el juego propuesto, se diseñó un cuestionario más de ocho afirmaciones con el fin de recopilar información de los cinco profesores participantes (véase Tabla 13). Como se mencionó anteriormente, este cuestionario sigue las recomendaciones de Bourgonjon et al., (2013) y Bodnar et al., (2015) para validar la aceptación y las expectativas de los profesores sobre el juego mediante la evaluación de cinco dimensiones: *complejidad* (i.e., el grado en el que juego serio es percibido como relativamente difícil de entender y usar), *experiencia* (i.e., la experiencia previa con los juegos serios puede aumentar la disposición de los profesores a utilizar uno nuevo en sus aulas), *oportunidades de aprendizaje* (i.e., los profesores pueden compartir la creencia de que un juego serio puede considerarse una herramienta alternativa para apoyar las teorías educativas), *innovación personal* en el ámbito de la tecnología de la información (i.e., la voluntad de los profesores de probar cualquier nueva tecnología de la información, como un juego serio), y *norma subjetiva* (i.e., la presión social percibida para usar o no el juego serio propuesto). Al igual que el cuestionario anterior, este cuestionario final utiliza una escala Likert de cinco puntos con las mismas alternativas de respuesta: “Totalmente en desacuerdo” (TD) = -2.0, “En desacuerdo” (ED) = -1.0, “Neutral” (N) = 0.0, “De acuerdo” (DA) = 1.0 y “Totalmente de acuerdo” (TA) = 2.0.

**Tabla 13.** Cuestionario para recoger las percepciones de los profesores (adaptado de Bourgonjon et al., (2013) y Bodnar et al., (2015))

Dimensión	Afirmaciones	TD	ED	N	DA	TA	M	S
Complejidad	1. Me siento cómodo(a) al saber que los estudiantes repasan los conceptos vistos en clase mientras juegan.							
	2. En mi opinión, el juego es relativamente fácil de entender y jugar.							
Experiencia	3. El juego no requiere demasiado tiempo para emplearse en cursos de Ingeniería de Software a nivel universitario.							
	4. Esta experiencia ha aumentado mi disposición a utilizar juegos en otros cursos.							
Oportunidades de aprendizaje	5. El juego es útil para relacionarse con los conocimientos fundamentales del estándar ISO/IEC/IEEE 26515:2018 a nivel de universitario.							
	6. El juego puede ayudar a lo(a)s estudiantes a comprender los tópicos teóricos relacionados con el desarrollo ágil de software y el desarrollo de la información de usuarios.							
Innovación personal	7. El juego puede aumentar la voluntad y apertura al cambio de otro(a)s profesores que no tienen experiencia previa con el uso de juegos serios como soporte educativo.							
Norma subjetiva	8. Sería correcto sugerirle a mis colegas el uso de este juego como herramienta alternativa de apoyo para sus cursos sobre Ingeniería de Software.							

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

### 4.3. Estrategia de evaluación

La evaluación correspondió con un caso de estudio que implicó a estudiantes y profesores del noveno semestre del programa de la Licenciatura de Ingeniería en Computación de la UTM. En este sentido, la estrategia que se siguió correspondió con los siguientes pasos:

1. Los estudiantes recibieron clases teóricas sobre dos tópicos: (1) Ingeniería de Software y desarrollo ágil de software y (2) Aspectos fundamentales del estándar ISO/IEC/IEEE 26515:2018. Las clases se llevaron a cabo diariamente durante cuatro semanas dentro del curso “Pruebas de software” en el tópico “Pruebas en entornos ágiles”, considerando a la evaluación de la información generada para los usuarios.
2. Posteriormente, los estudiantes fueron divididos aleatoriamente en dos grupos: un grupo de control y otro experimental. Dicha división no distinguió o evaluó variaciones entre el desempeño de mujeres y/o hombres. En este sentido, se buscó establecer una base comparativa entre los resultados obtenidos por el grupo de control al reforzar los conocimientos aprendidos en el curso siguiendo un enfoque tradicional (i.e., clase teórica, repaso autodidacta, examen teórico), y los obtenidos por el grupo experimental al utilizar “*Serpientes y escaleras: Un juego de soporte al aprendizaje del estándar ISO/IEC/IEEE 26515:2018*” para lograr el mismo fin.

3. Al concluir las clases teóricas, los estudiantes del grupo de control respondieron un examen teórico con el fin de evaluar su comprensión de los temas aprendidos en el aula y, por ende, determinar su rendimiento académico. Por otro lado, los estudiantes del grupo experimental participaron en cinco sesiones de dos horas durante las cuales se organizaron competencias entre parejas hasta llegar a una final entre dos estudiantes para cada sesión de juego, con el objetivo de consolidar su comprensión de los conceptos teóricos. Al finalizar las sesiones de juego, los estudiantes del grupo experimental respondieron el mismo examen teórico que se aplicó a su contraparte del grupo de control.
4. Al terminar el examen teórico, los estudiantes del grupo experimental respondieron un cuestionario diseñado para evaluar el rendimiento del juego propuesto en esta tesis, de acuerdo con las dimensiones definidas por el MEEGA+ (Petri et al., 2019).
5. Con el fin de concluir la evaluación empírica, se realizó una tercera evaluación considerando el cuestionario resultante de considerar las recomendaciones de Bourgonjon et al. (2013) y Bodnar et al., (2015) para evaluar la percepción de los(as) profesores(as) sobre el juego.
6. Finalmente, los datos recopilados con las evaluaciones anteriores fueron sometidos a un análisis estadístico descriptivo con el fin de validar o refutar la hipótesis planteada en la tesis.

#### 4.4. Resultados de la evaluación

Considerando la estrategia descrita anteriormente para el caso de estudio, ambos grupos de estudiantes tomaron clases teóricas en un formato presencial tradicional con el propósito de reafirmar los conceptos fundamentales relacionados con la Ingeniería de Software y los métodos ágiles, el estándar ISO/IEC/IEEE 26515:2018, y el desarrollo de la documentación en entornos ágiles de desarrollo. Las clases se llevaron a cabo en el Laboratorio de Metodologías de Desarrollo e Inteligencia Artificial de la División de Estudios de Posgrado de la UTM a través de sesiones presenciales diarias de una hora durante cuatro semanas (véase Figura 4.1).



**Figura 4.1.** Estudiantes tomando las clases teóricas como parte de la evaluación empírica del juego

Durante este periodo de tiempo se pudo observar que, durante los primeros 20 a 25 minutos, lo(a)s estudiantes estuvieron enfocados en la explicación que se les daba de los temas y respondieron

con ánimo a las preguntas planteadas en momentos claves de la presentación. Sin embargo, después de este tiempo, el nivel de atención y enfoque comenzó a disminuir y alguno(a)s estudiantes comenzaron a distraerse, mostrando signos de aburrimiento. Se asume que es normal que esto ocurra en una clase teórica y que, por consecuencia, se espera que lo(a)s estudiantes repasen los conceptos previos a una evaluación. Tras concluir las clases teóricas, los integrantes del grupo de control tomaron cinco días más para repasar sus apuntes con el fin de resolver un examen de conocimientos que buscó evaluar su comprensión de los conceptos presentados en el aula. El análisis de sus respuestas dio como resultado la creación de la Figura 4.2(a).



**Figura 4.2.** Resultados de la evaluación sobre el grupo de control mediante un examen de conocimientos

Como se aprecia en el lado izquierdo de esta figura, el 39% de las respuestas proporcionadas por lo(a)s estudiantes a las 30 preguntas mostradas en la Tabla 11, fueron correctas, mientras que el 61.0% de las respuestas fueron incorrectas. En consecuencia, la calificación promedio del grupo de control fue de 6.1 sobre 10. Es relevante destacar que ninguno de los estudiantes en el grupo de control logró responder correctamente las 30 preguntas de la evaluación. Considerando un enfoque tradicional de enseñanza, a no ser por la realización de prácticas y/o proyectos, el proceso anterior definiría la calificación de un estudiante para, por ejemplo, una evaluación parcial.

Por otro lado, la evaluación realizada con el grupo experimental difirió de acuerdo con la estrategia definida en la sección anterior del documento. En este sentido, una vez que se terminaron las clases teóricas, los estudiantes que conformaron el grupo experimental acudieron al mismo laboratorio durante cinco días para realizar sesiones de juego de dos horas (véase Figura 4.3). El proceso consistió en crear una competencia cada media hora entre los estudiantes. Por lo tanto, se formaron parejas aleatorias de estudiantes para jugar “*Serpientes y escaleras: Un juego de soporte al aprendizaje del estándar ISO/IEC/IEEE 26515:2018*” y se informó a los estudiantes que cada batalla tendría una duración máxima de 10 minutos. Con el objetivo de promover la participación, el estudiante que acumuló más victorias durante las sesiones de juego recibió una pizza como premio por su esfuerzo.

El proceso comenzó con los estudiantes haciendo doble clic en el icono para abrir el juego, que había sido previamente instalado en las computadoras del laboratorio, y acceder al menú de inicio del juego (véase Figura 3.10). Cabe mencionar que los estudiantes se mostraron interesados por comenzar a jugar, argumentando que la apariencia del juego les parecía atractiva a la vista. La mayoría de lo(a)s estudiantes ya conocían con anterioridad la dinámica del juego de “Serpientes y escaleras”, por lo que

les fue fácil entender las reglas y, además, encontraron interesante que esta versión digital del juego incluyera una trivía.



**Figura 4.3.** Estudiantes del grupo experimental relacionándose con el juego

De manera similar, cuando lo(a)s estudiantes se percataron que podían elegir distintos mapas y colores para su avatar, se hicieron evidentes la atracción e intriga por el juego, independientemente si se tenía experiencia previa con los videojuegos o no. Otro aspecto que es importante mencionar, es que lo(a)s estudiantes se centraron en responder correctamente las preguntas para avanzar en el tablero, sin mostrar interés en los elementos que se encontraban en éste. Es decir, se observó que la mayoría no tuvo la curiosidad de probar el botón de pausa que estuvo siempre presente durante la sesión de juego. Se pudo notar también que, aproximadamente a los cinco minutos de haber iniciado la primera sesión de juego, lo(a)s estudiantes comenzaron a familiarizarse con el juego, haciendo su interacción con el mismo más eficiente. Por otro lado, no se observó a estudiantes frustrados por no entender qué debían hacer durante el juego, lo que indica que la dinámica del juego “Serpientes y escaleras” facilita la jugabilidad y no interfiere con la comprensión de conceptos teóricos.

Al concluir estas sesiones de juego, los estudiantes del grupo experimental resolvieron el mismo examen teórico que se les aplicó a sus compañero(a)s del grupo de control con el objetivo de encontrar variaciones en su desempeño. Como se puede observar, los resultados obtenidos variaron significativamente de un 39.0% de respuestas correctas para el grupo de control (véase Figura 4.2) a un 82.0% para el grupo experimental (véase Figura 4.3). En este sentido, es importante considerar lo siguiente: (1) se desconoce si los estudiantes del grupo experimental se encontraban más motivados que los del grupo de control al responder las preguntas del examen, como consecuencia de haber participado en las sesiones de juego, (2) tampoco se puede afirmar que los estudiantes del grupo de control sean menos inteligentes que su contraparte del grupo experimental, y (3) se reconoce que los estudiantes del grupo experimental respondieron el examen de conocimientos inmediatamente después de haber concluido la última sesión de juego, mientras que los estudiantes del grupo de control tuvieron cinco días para repasar los conceptos teóricos aprendidos en el aula antes de resolver el mismo examen y, por lo tanto, se asumió que, bajo un enfoque tradicional, este tiempo fue destinado para repasar sus apuntes sobre el curso.



**Figura 4.4.** Resultados de la evaluación sobre el grupo experimental mediante un examen de conocimientos

Posteriormente, los estudiantes del grupo de control proporcionaron sus percepciones sobre “*Serpientes y escaleras: Un juego de soporte al aprendizaje del estándar ISO/IEC/IEEE 26515:2018*” al responder las preguntas planteadas en el instrumento de evaluación de MEEGA+ (Petri et al., 2019). Como se mencionó anteriormente, dicho instrumento facilita la evaluación de ocho dimensiones que fueron propuestas para determinar la efectividad de juegos serios, tanto digitales como los no digitales, en un contexto educativo. En este sentido, la usabilidad está determinada por cinco subdimensiones que se definen de la siguiente manera:

- **Estética:** Se evalúa si la interfaz del juego permite una interacción agradable y satisfactoria con el usuario.
- **Capacidad de aprendizaje:** Se evalúa si el juego permite a los usuarios aprender a jugarlo fácil y rápidamente.
- **Operabilidad:** Se evalúa el grado en que el juego cumple con atributos que facilitan su operación y control.
- **Accesibilidad:** Se evalúa si el juego puede ser utilizado por personas con discapacidad visual baja/moderada y/o daltonismo.
- **Protección de errores:** Se evalúa si el juego protege a los usuarios de cometer errores, una característica que solamente aplica en la evaluación de juegos digitales.

La Tabla 14 resume las respuestas de los estudiantes del grupo experimental considerando las preguntas relacionadas con la dimensión de *usabilidad*. Como se podrá observar en la Figura 4.5, dicha dimensión fue evaluada positivamente puesto que, en general, todas las respuestas se mantuvieron entre los valores de “De acuerdo” (DA) y “Totalmente de acuerdo” (TA). Con esta evaluación destacan de forma más específica las subdimensiones de *capacidad de aprendizaje* y *protección de errores* con los valores para ambos casos de *M* entre 1.9 y 2.0 y *S* entre 0.0 y 0.3 (véanse las filas de la Tabla 14 resaltadas en color verde). Por otro lado, se presentaron pequeñas disparidades para las subdimensiones de *estética* y *accesibilidad*, puesto que el 20% de los estudiantes indicó que era necesario incluir un tutorial sobre el juego puesto que se sintieron despistados al momento de comenzar a jugar (3. *Considero que no se requiere de un tutorial para comprender correctamente la temática del juego*,  $M = 0.9$ ,  $S = 1.0$ ) y que la apariencia del juego no se pudo ajustar a sus preferencias

(10. La apariencia del juego (fuentes, colores, etc.) se puede personalizar de acuerdo con las preferencias de un(a) jugador(a),  $M = 0.7$ ,  $S = 1.2$ ). Ambos resultados se resaltan de color amarillo en la Tabla 14. Con el fin de indagar más sobre la razón por la cual estos estudiantes respondieron de esta manera, se realizó una entrevista que condujo a determinar dos razones principales: (1) dos participantes afirmaron no tener relación alguna con los videojuegos puesto que siempre los han considerado un objeto distractor o para malgastar el tiempo; por lo tanto, existía una animadversión al momento de realizar las sesiones de juego que pudo haber afectado la aceptación definitiva del mismo y (2) otros dos participantes consideraron que cambiar el color del avatar, el tipo de dado, los contenidos temáticos, el mapa y la complejidad del juego no era suficiente como para considerar una personalización que correspondiera con sus gustos. Es importante recordar que, a pesar de estas respuestas negativas, el 100% de los estudiantes reconoció que el juego propuesto fue fácil de jugar (*capacidad de aprendizaje*) y que, al mismo tiempo, la forma en que fue diseñado impide que el(la) jugador(a) cometa errores durante las sesiones de juego (*protección de errores*). No obstante, aun cuando solamente el 20% de los participantes no estuvieron satisfechos con no contar con un tutorial o la personalización del juego, representadas por las dimensiones de *estética* y *accesibilidad*, los comentarios vertidos por los estudiantes aportaron información valiosa para mejorar el juego, de tal forma que se agregará, en un futuro, un capítulo inicial que facilite la relación inicial entre jugador y juego, previo a comenzar la competencia. Aunado a esto, se agregará una mecánica más de concurso que permita que el jugador ganador acumule puntos que podrá canjear en una tienda virtual de accesorios, a modo de recompensa, con el fin de personalizar su *cubitante* (e.g., lentes de sol, sombreros, tenis, accesorios varios).

**Tabla 14.** Percepciones de los estudiantes sobre la dimensión de usabilidad con MEEGA+

Dimensión		Afirmaciones	TD	ED	N	DA	TA	M	S
Usabilidad	Estética	1. El diseño del juego es atractivo con relación a su interfaz, gráficos, etc.	0.0	0.0	0.0	3.0	7.0	1.7	0.5
		2. Las fuentes de texto y los colores usados en el juego me parecen adecuados y consistentes.	0.0	0.0	0.0	9.0	1.0	1.1	0.3
		3. Considero que no se requiere de un tutorial para comprender correctamente la temática del juego.	0.0	2.0	0.0	5.0	3.0	0.9	1.0
	Capacidad de aprendizaje	4. La jugabilidad se puede interpretar fácilmente.	0.0	0.0	0.0	1.0	9.0	1.9	0.3
		5. La mayoría de las personas podrían aprender rápidamente los comandos del juego.	0.0	0.0	0.0	0.0	10.0	2.0	0.0
	Operabilidad	6. El juego puede jugarse fácilmente.	0.0	0.0	0.0	3.0	7.0	1.7	0.5
		7. Las reglas del juego son claras y concisas.	0.0	0.0	0.0	0.0	10.0	2.0	0.0
	Accesibilidad	8. El tamaño y estilo de las fuentes usadas en el juego facilitan la lectura de los textos.	0.0	0.0	0.0	7.0	3.0	1.3	0.5
		9. Los colores usados son los adecuados para la temática del juego.	0.0	0.0	0.0	2.0	8.0	1.8	0.4
		10. La apariencia del juego (fuentes, colores, etc.) se puede personalizar de acuerdo con las preferencias de un(a) jugador(a).	1.0	1.0	0.0	6.0	2.0	0.7	1.2
	Protección de errores	11. El juego impide que el(la) jugador(a) cometa errores.	0.0	0.0	0.0	1.0	9.0	1.9	0.3

Dimensión	Afirmaciones	TD	ED	N	DA	TA	M	S
	12. Es fácil recuperarse de errores cometidos en el juego.	0.0	0.0	0.0	0.0	10.0	2.0	0.0

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

Con relación a la dimensión de *confianza*, la Tabla 12 muestra que el 100% de los estudiantes del grupo experimental confirmó que el juego fue fácil de jugar (13. Desde el inicio del juego da la impresión de que es fácil de jugar, M = 1.4, S = 0.5) y que, además, les permitió progresar, como resultado de su esfuerzo, en el aprendizaje de los temas relacionados con los métodos ágiles y el proceso de documentación de la información descrito por el estándar ISO/IEC/IEEE 26515:2018 (14. Tanto el contenido como la estructura del juego dan al(a) jugador(a) la confianza de que aprenderá algo nuevo, M = 1.4, S = 0.5). La Figura 4.6 muestra las respuestas de los participantes.

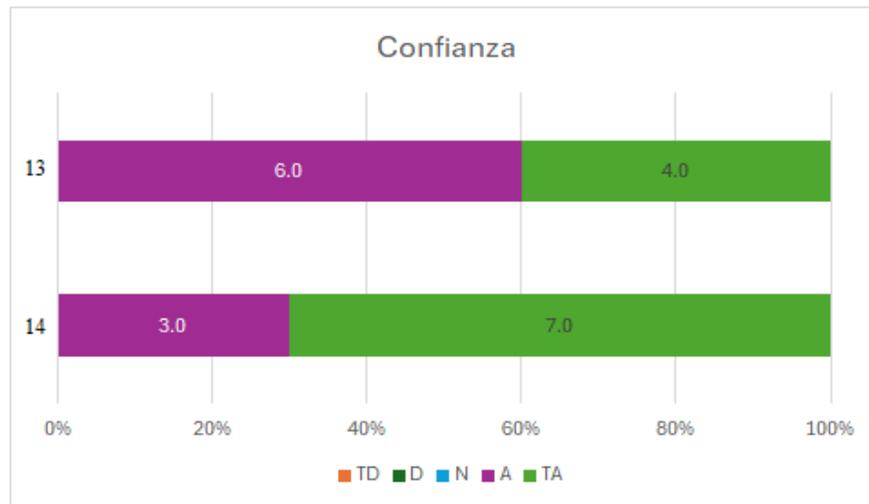


Figura 4.5. Evaluación de la dimensión de usabilidad en el juego propuesto

**Tabla 15.** Percepciones de los estudiantes sobre la dimensión de confianza con MEEGA+

Dimensión	Afirmaciones	TD	ED	N	DA	TA	M	S
Confianza	13. Desde el inicio del juego da la impresión de que es fácil de jugar.	0.0	0.0	0.0	6.0	4.0	1.4	0.5
	14. Tanto el contenido como la estructura del juego dan al(a) jugador(a) la confianza de que aprenderá algo nuevo.	0.0	0.0	0.0	3.0	7.0	1.7	0.5

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

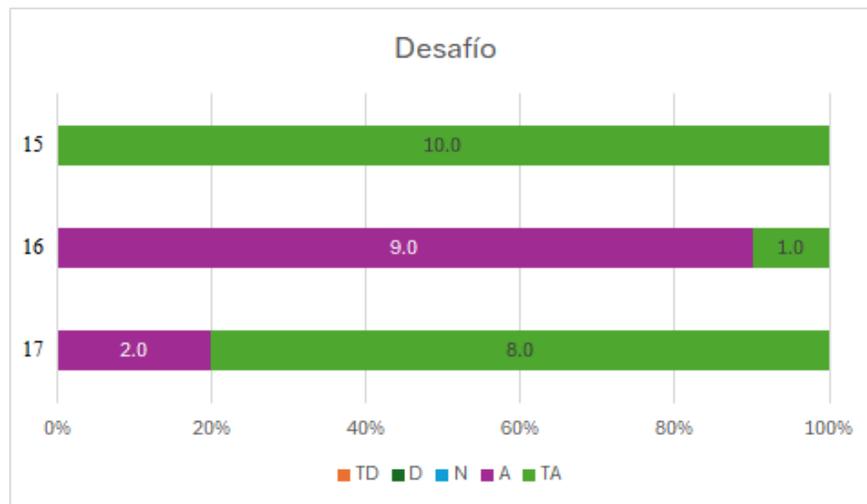
**Figura 4.6.** Evaluación de la dimensión de confianza en el juego propuesto

Resultados similares se obtuvieron para la dimensión de *desafío* puesto que las respuestas de todo(a)s los participantes se mantuvieron entre los valores de 1.1 a 2.0 para *M* con desviaciones estándares bajas entre 0.0 y 0.4, lo que evidencia un alto nivel de consenso entre las opiniones proporcionadas por ésto(a)s. Cabe destacar que, como lo muestra la fila de la Tabla 16 resaltada en color verde, el 100% de lo(a)s estudiantes consideró que los niveles del juego son lo suficientemente complicados como para promover el aprendizaje del estándar ISO/IEC/IEEE 26515:2018 (véase la Figura 4.7).

**Tabla 16.** Percepciones de los estudiantes sobre la dimensión de desafío con MEEGA+

Dimensión	Afirmaciones	TD	ED	N	DA	TA	M	S
Desafío	15. La dificultad del juego es la apropiada.	0.0	0.0	0.0	0.0	10.0	2.0	0.0
	16. El ritmo de las misiones (situaciones, variaciones, nuevos obstáculos, etc.) varía adecuadamente en el juego conforme el tiempo avanza.	0.0	0.0	0.0	9.0	1.0	1.1	0.3
	17. El juego y las misiones no se vuelven monótonas o aburridas a medida que avanza la historia.	0.0	0.0	0.0	2.0	8.0	1.8	0.4

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



**Figura 4.7.** Evaluación de la dimensión de desafío en el juego propuesto

En cuanto a la *satisfacción* de los estudiantes destacan dos percepciones. En primera instancia, la fila de la Tabla 17 resaltada en color verde muestra que el 100% de los estudiantes consideró, sin duda alguna, que otros estudiantes podrían relacionarse con los conceptos fundamentales sobre el estándar ISO/IEC/IEEE 26515:2018 a través del esfuerzo puesto en el juego (21. *Recomendaría este juego a mis amigo(a)s o compañero(a)s*),  $M = 2.0$ ,  $S = 0.0$ ). Por otro lado, se obtuvo una pequeña discrepancia sobre el aprendizaje percibido en el juego (20. *El juego genera satisfacción con el aprendizaje obtenido*,  $M = 1.1$ ,  $S = 0.5$ ), puesto que uno de los estudiantes argumentó, en la entrevista, que no estaba satisfecho con el aprendizaje obtenido (como lo muestra la fila de la Tabla 17 resaltada en color amarillo). En este sentido, es importante mencionar que este estudiante era uno de los dos que cursaban las materias optativas de “Inteligencia Artificial” y que también fue uno de los participantes que no estaba relacionado con los videojuegos, por lo que, posiblemente, esto pudo influenciar su opinión respecto a esta dimensión. Los resultados se muestran en la Figura 4.8.

**Tabla 17.** Percepciones de los estudiantes sobre la dimensión de satisfacción con MEEGA+

Dimensión	Afirmaciones	TD	ED	N	DA	TA	M	S
Satisfacción	18. El completar cada una de las misiones del juego ofrece una sensación de logro.	0.0	0.0	0.0	10.0	0.0	1.0	0.0
	19. Con esfuerzo es posible sortear los obstáculos del juego e ir avanzando.	0.0	0.0	0.0	5.0	5.0	1.5	0.5
	20. El juego genera satisfacción con el aprendizaje obtenido.	0.0	0.0	1.0	7.0	2.0	1.1	0.5
	21. Recomendaría este juego a mis amigo(a)s o compañero(a)s.	0.0	0.0	0.0	0.0	10.0	2.0	0.0

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

La mayor discrepancia de toda la medición con MEEGA+ se obtuvo para la dimensión de la *diversión*, puesto que, como se observa en la fila resaltada en color verde en la Tabla 18, si bien el 100% de los estudiantes consideró que el juego en sí fue divertido (22. *Las personas pueden divertirse*

con el juego,  $M = 1.4$ ,  $S = 0.5$ ), el 60% permaneció neutral (véase la fila resaltada en color amarillo en la Tabla 18) cuando se le preguntó si había experimentado algún tipo de felicidad durante su interacción con el juego a lo largo de las competencias (23. Algunos elementos del juego (diálogos, misiones, imágenes, etc.) pueden hacer que lo(a)s jugadore(a)s se ríen,  $M = 0.5$ ,  $S = 0.7$ ). Esto tiene sentido, puesto que, en primer lugar, el juego no se diseñó considerando cinemáticas que pudieran generar risas en los estudiantes y, en segundo lugar, dado que se trata de un juego de trivia, se comprobó en las entrevistas que la dinámica de lanzar dados y responder preguntas generó aburrimiento en estos estudiantes a largo plazo (véase Figura 4.9).

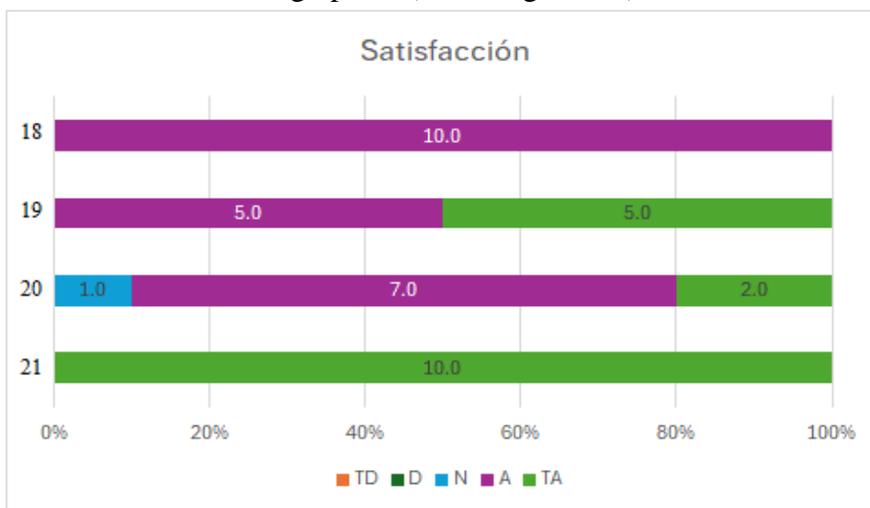


Figura 4.8. Evaluación de la dimensión de satisfacción en el juego propuesto

Tabla 18. Percepciones de los estudiantes sobre la dimensión de diversión con MEEGA+

Dimensión	Afirmaciones	TD	D	N	DA	TA	M	S
Diversión	22. Las personas pueden divertirse con el juego.	0.0	0.0	0.0	6.0	4.0	1.4	0.5
	23. Algunos elementos del juego (diálogos, misiones, imágenes, etc.) pueden hacer que lo(a)s jugadore(a)s se ríen.	0.0	0.0	6.0	3.0	1.0	0.5	0.7

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



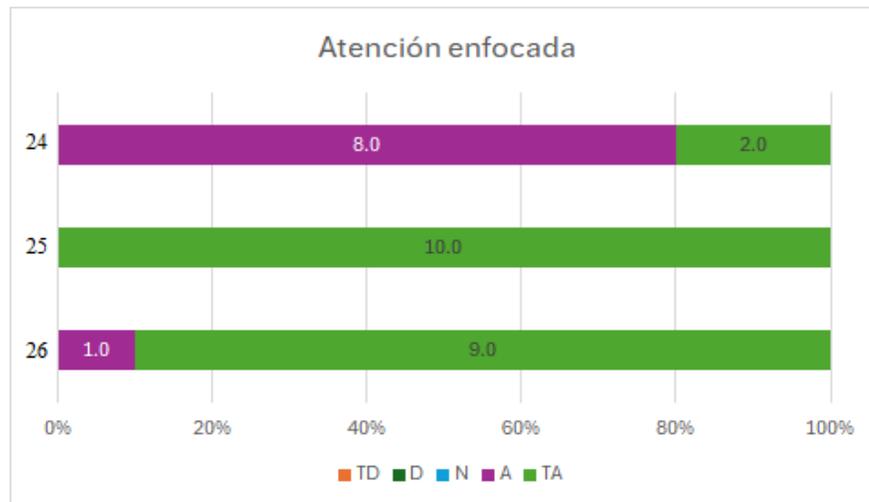
**Figura 4.9.** Evaluación de la dimensión de diversión en el juego propuesto

La evaluación de la dimensión de la *atención enfocada* obtuvo resultados positivos, sin discrepancia en las percepciones de lo(a)s estudiantes. En este sentido, el 100% de lo(a)s participantes estuvo de acuerdo con ligeras variaciones en que el diseño del juego los ayudó a mantenerse atentos (33. *Hay aspectos del juego que pueden llamar la atención del(a) jugador(a) desde el inicio*,  $M = 1.2$ ,  $S = 0.4$ ) y que éste es lo suficientemente inmersivo para no distraerse durante las competencias (35. *Es posible que un(a) jugador(a) se olvide de su entorno mientras juega*,  $M = 1.9$ ,  $S = 0.3$ ), lo que indica un alto grado de aceptación. Además, no se presentó variación alguna en sus respuestas cuando se les preguntó sobre el nivel de concentración experimentado durante las sesiones de juego (34. *La concentración en las misiones del juego hace que se pierda la noción del tiempo*,  $M = 2.0$ ,  $S = 0.0$ ) (véase fila de la Tabla 19 resaltada en color verde). Esto es obvio, puesto que, como se mencionó anteriormente, lo(a)s estudiantes deben estar atentos en todo momento al lanzamiento del dado, las casillas que su avatar se desplaza, la trivía, y la selección de la respuesta correcta. Las percepciones de lo(a)s estudiantes se resumen en la Figura 4.10.

**Tabla 19.** Percepciones de los estudiantes sobre la dimensión de atención enfocada con MEEGA+

Dimensión	Afirmaciones	TD	D	N	DA	TA	M	S
Atención enfocada	33. Hay aspectos del juego que pueden llamar la atención del(a) jugador(a) desde el inicio.	0.0	0.0	0.0	8.0	2.0	1.2	0.4
	34. La concentración en las misiones del juego hace que se pierda la noción del tiempo.	0.0	0.0	0.0	0.0	10.0	2.0	0.0
	35. Es posible que un(a) jugador(a) se olvide de su entorno mientras juega.	0.0	0.0	0.0	1.0	9.0	1.9	0.3

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



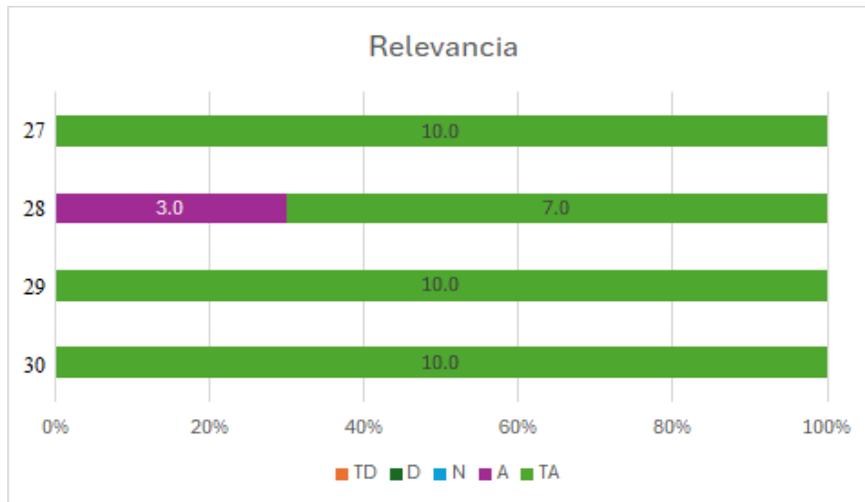
**Figura 4.10.** Evaluación de la dimensión de atención enfocada en el juego propuesto

La *relevancia* fue la dimensión mejor evaluada por lo(a)s estudiantes puesto que se obtuvieron calificaciones perfectas ( $M = 2.0$  y  $S = 0.0$ ) para las tres afirmaciones resaltadas en la Tabla 20, las cuales representan a la mayoría en esta categoría de la evaluación (véase Figura 4.11). Con esta información, se demuestra que lo(a)s participantes consideran que el juego propuesto contribuye a que alcancen sus objetivos (i.e., reforzar sus conocimientos teóricos sobre el estándar ISO/IEC/IEEE 26515:2018) y adquieran conocimiento que les puede ser útil en su futuro profesional.

**Tabla 20.** Percepciones de los estudiantes sobre la dimensión de relevancia con MEEGA+

Dimensión	Afirmaciones	TD	ED	N	DA	TA	M	S
Relevancia	36. Los contenidos mostrados en el juego son de relevancia para la clase teórica.	0.0	0.0	0.0	0.0	10.0	2.0	0.0
	37. Existe una relación clara entre el juego y los objetivos de aprendizaje.	0.0	0.0	0.0	3.0	7.0	1.7	0.5
	38. El juego es una forma adecuada de fortalecer el aprendizaje sobre el tema.	0.0	0.0	0.0	0.0	10.0	2.0	0.0
	39. Es posible que los estudiantes prefieran reforzar sus conocimientos con este juego que con otras formas tradicionales (libros, diapositivas, etc.).	0.0	0.0	0.0	0.0	10.0	2.0	0.0

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



**Figura 4.11.** Evaluación de la dimensión de relevancia en el juego propuesto

Finalmente, la evaluación de la dimensión del *aprendizaje percibido* se mantuvo del lado positivo de la escala (-2.0 a 2.0), destacando el consenso de las opiniones vertidas por lo(a)s estudiantes sobre el juego (41. *La enseñanza brindada por el juego fue más eficiente para el aprendizaje de los tópicos vistos en clase que otros métodos tradicionales*,  $M = 2.0$ ,  $S = 0.0$ ) (véase fila de la Tabla 21 resaltada en color verde). Estas opiniones confirman, además, que lo(a)s estudiantes percibieron que el juego facilita el aprendizaje del estándar ISO/IEC/IEEE 26515:2018.

**Tabla 21.** Percepciones de los estudiantes sobre la dimensión de aprendizaje percibido con MEEGA+

Dimensión	Afirmaciones	TD	ED	N	DA	TA	M	S
Aprendizaje percibido	40. El juego puede contribuir a mejorar el aprendizaje de los tópicos vistos en la clase teórica.	0.0	0.0	0.0	4.0	6.0	1.6	0.5
	41. La enseñanza brindada por el juego fue más eficiente para el aprendizaje de los tópicos vistos en clase que otros métodos tradicionales.	0.0	0.0	0.0	0.0	10.0	2.0	0.0

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

La Figura 4.12 ilustra el consenso en las percepciones de lo(a)s estudiantes sobre el juego propuesto como una alternativa de soporte educativo a nivel universitario.

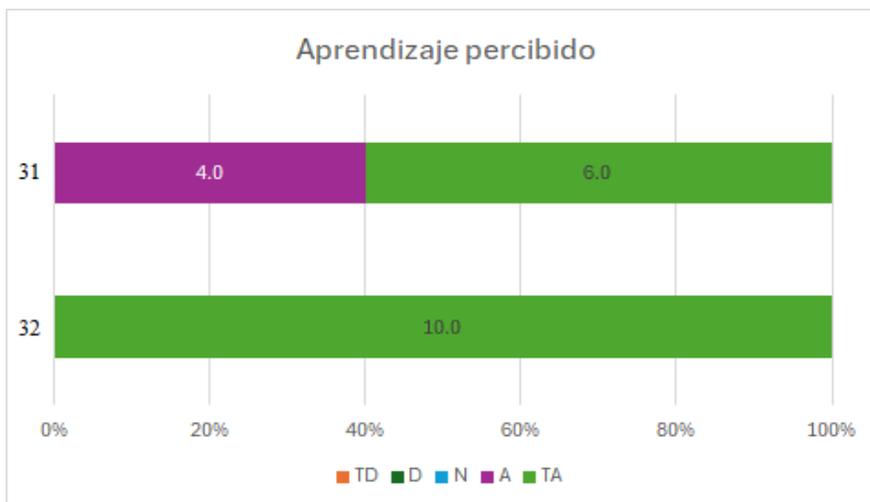


Figura 4.12. Evaluación de la dimensión de aprendizaje percibido en el juego propuesto

Por otro lado, considerando la evaluación de las dimensiones de *complejidad*, *experiencia*, *oportunidades de aprendizaje*, *innovación personal*, y *norma subjetiva*, sugeridas por Bourgonjon et al., (2013) y Bodnar et al., (2015) para determinar la aceptación y expectativas de lo(a)s profesores sobre un juego serio, las percepciones de lo(a)s participantes fueron bastante positivas (véase Figura 4.13). La información recopilada destaca dos casos puntuales. En primera instancia, referente a la dimensión de *complejidad*, mientras que el 100% de lo(a)s profesores consideró que el juego era de fácil uso para lograr su cometido de apoyo a la enseñanza (2. *En mi opinión, el juego es relativamente fácil de entender y jugar*,  $M = 2.0$ ,  $S = 0.0$ ), el 20% no se sintió cómodo al considerarlo como una herramienta alternativa para reforzar el aprendizaje de sus estudiantes (1. *Me siento cómodo(a) al saber que los estudiantes repasan los conceptos vistos en clase mientras juegan*,  $M = 1.0$ ,  $S = 0.6$ ).

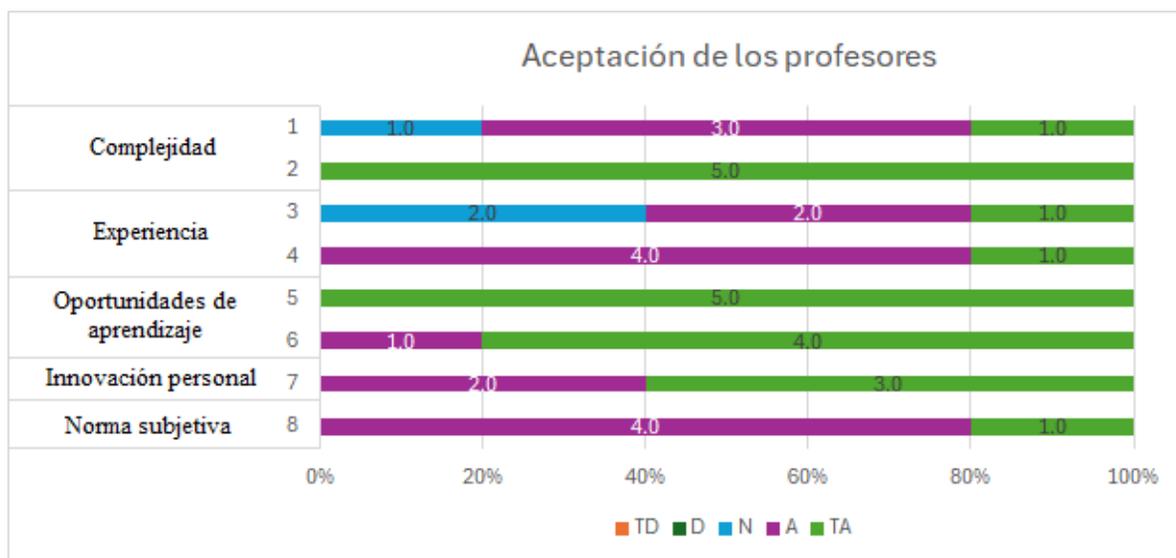


Figura 4.13. Percepciones de lo(a)s profesores sobre el juego propuesto

Estas opiniones se pueden observar con las filas de Tabla 22 resaltadas en color verde y amarillo, respectivamente. Al indagar sobre su respuesta con una entrevista, el profesor afirmó no tener la

experiencia requerida para introducir este tipo de tecnología en sus cursos y su preocupación principal se relacionaba con la manera en la que debería compaginar el juego con los temas teóricos. Si bien este problema es bastante recurrente cuando se introducen juegos serios a clases reales, la aceptación de estas herramientas educativas por parte de lo(a)s profesores ha sido extensamente analizada en investigaciones como las de Sandí-Delgado et al., (2022) y Valencia y Duque (2023), que han generado también estrategias para el desarrollo de competencias con el fin de apoyar a lo(a)s profesores en la realización de una adopción controlada.

El segundo caso por resaltar es el de la dimensión de *experiencia*, puesto que el 40% de los profesores tuvieron una opinión neutral cuando se les preguntó sobre la introducción del juego en cursos reales (3. *El juego no requiere demasiado tiempo para emplearse en cursos de Ingeniería de Software a nivel universitario*,  $M = 0.8$ ,  $S = 0.7$ ). Al indagar sobre su respuesta, se determinó que los profesores desconocen si este tipo de recursos educativos sean aceptados en universidades del SUNEI como instrumentos de práctica o soporte a cursos reales.

**Tabla 22.** Percepciones de los profesores sobre el juego propuesto

Dimensión	Afirmaciones	TD	ED	N	DA	TA	M	S
Complejidad	1. Me siento cómodo(a) al saber que los estudiantes repasan los conceptos vistos en clase mientras juegan.	0.0	0.0	1.0	3.0	1.0	1.0	0.6
	2. En mi opinión, el juego es relativamente fácil de entender y jugar.	0.0	0.0	0.0	0.0	5.0	2.0	0.0
Experiencia	3. El juego no requiere demasiado tiempo para emplearse en cursos de Ingeniería de Software a nivel universitario.	0.0	0.0	2.0	2.0	1.0	0.8	0.7
	4. Esta experiencia ha aumentado mi disposición a utilizar juegos en otros cursos.	0.0	0.0	0.0	4.0	1.0	1.2	0.4
Oportunidades de aprendizaje	5. El juego es útil para relacionarse con los conocimientos fundamentales del estándar ISO/IEC/IEEE 26515:2018 a nivel de universitario.	0.0	0.0	0.0	0.0	5.0	2.0	0.0
	6. El juego puede ayudar a lo(a)s estudiantes a comprender los tópicos teóricos relacionados con el desarrollo ágil de software y el desarrollo de la información de usuarios.	0.0	0.0	0.0	1.0	4.0	1.8	0.4
Innovación personal	7. El juego puede aumentar la voluntad y apertura al cambio de otro(a)s profesores que no tienen experiencia previa con el uso de juegos serios como soporte educativo.	0.0	0.0	0.0	2.0	3.0	1.6	0.5
Norma subjetiva	8. Sería correcto sugerirle a mis colegas el uso de este juego como herramienta alternativa de apoyo para sus cursos sobre Ingeniería de Software.	0.0	0.0	0.0	4.0	1.0	1.2	0.4

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

De hecho, estos profesores consideran que sería necesario que primero, y desde la jefatura de carrera, se plantee el uso de estos juegos serios dentro de los cursos reales puesto que su uso pudiera generar opiniones encontradas entre lo(a)s estudiantes. Sin embargo, se considera que el uso de los juegos serios puede ser una actividad de libre cátedra para cualquier profesor(a), independientemente

de que se trate o no de una institución académica del SUNEQ y, en este escenario, la disposición para su uso es más importante que otra cosa.

Finalmente, las dimensiones de *oportunidades de aprendizaje, innovación personal, y norma subjetiva* fueron evaluadas positivamente por lo(a)s profesores. De hecho, independientemente de los desacuerdos sobre usar o no el juego en los cursos reales, el 100% de los profesores consideró que “*Serpientes y escaleras: Un juego de soporte al aprendizaje del estándar ISO/IEC/IEEE 26515:2018*” es indudablemente un soporte alternativo para relacionarse con el estándar analizado (5. *El juego es útil para relacionarse con los conocimientos fundamentales del estándar ISO/IEC/IEEE 26515:2018 a nivel de universitario, M = 2.0, S = 0.0*).

#### 4.5. Conclusiones sobre los resultados de la evaluación

Como se puede observar en la sección anterior, el juego propuesto en esta tesis evidenció una *mejora notable en el rendimiento académico de los estudiantes*. Es decir, el grupo experimental, que utilizó el juego como herramienta de aprendizaje, obtuvo un 82% de respuestas correctas en el examen de evaluación, en contraste con el 39% de respuestas obtenidas por el grupo de control que no utilizó el juego. Esto sugiere que dicho juego ayudó a reforzar los conceptos teóricos de manera más efectiva que un enfoque tradicional.

Aunado a lo anterior, *lo(a)s estudiantes mostraron altos niveles de interés y motivación* durante las sesiones de juego. En este sentido, se puede afirmar que la dinámica competitiva establecida por el juego, la cual es apoyada por elementos visuales atractivos y la posibilidad de personalizar los avatares, *contribuyó además a mantener la atención y compromiso de lo(a)s estudiantes*. Sin embargo, también se identificó la necesidad de mejorar la introducción al juego, puesto que lo(a)s estudiantes sugirieron la incorporación de un tutorial que los ayude a familiarizarse rápidamente con las reglas y objetivos del juego.

Por otro lado, lo(a)s estudiantes del grupo experimental evaluaron positivamente la usabilidad del juego, *destacando la capacidad de aprendizaje y protección contra errores al jugar*. A pesar de algunas críticas menores sobre la estética y la accesibilidad, la mayoría de esto(a)s estudiantes encontró que el juego es fácil de usar y resulta efectivo para promover el aprendizaje sobre el estándar. Es verdad también que las opiniones variaron ligeramente en cuanto a la satisfacción con el aprendizaje percibido, debido a que alguno(a)s estudiantes mostraron reservas sobre la profundidad del conocimiento adquirido.

Además de estas percepciones de lo(a)s estudiantes, *lo(a)s profesore(a)s también indicaron que el juego propuesto puede ser un recurso educativo valioso para sus cursos*. Sin embargo, expresaron ciertas dudas sobre su posible integración a un plan formal de estudios en sus respectivas instituciones, debido principalmente a la falta de autorización y, en algunos casos, la escasa familiaridad con los juegos serios como herramienta de enseñanza. A pesar de estas preocupaciones, lo(a)s profesore(a)s reconocieron el potencial del juego para facilitar la comprensión de conceptos complejos del estándar de una manera interactiva y atractiva.

Finalmente, considerando la evaluación empírica presentada en la sección anterior de este documento, se ha recopilado información suficiente para aceptar la hipótesis planteada en el Capítulo 1 bajo las particularidades del caso de estudio que se ha diseñado y conducido, la cual se enuncia de la siguiente manera:

“*El uso de un juego serio en el aprendizaje del estándar ISO/IEC/IEEE 26515:2018 Systems and software engineering —Developing information for users in an agile environment, mejorará la motivación, satisfacción, y el desempeño de estudiantes de nivel licenciatura*”.

Es importante mencionar que los resultados y las aseveraciones aquí mostradas no pueden generalizarse para cualquier uso del juego presentado, puesto que esto requeriría el diseño y realización de una mayor cantidad de casos de estudio dentro de contextos educativos reales.

## 5. Conclusiones

En este último capítulo de la tesis se presentan las principales conclusiones derivadas de la investigación. Aunado a esto, se pretende también dar un cierre a este documento de tesis con el fin de resaltar los resultados más importantes que se hayan obtenido a través de la evaluación empírica.

### 5.1. Conclusiones derivadas de la investigación

La documentación en los métodos ágiles es una tarea esencial que, aunque a menudo se subestima, desempeña un papel crucial en la eficiencia y el éxito a largo plazo de los proyectos de software. La percepción común de que los métodos ágiles priorizan la entrega de software funcional sobre la documentación detallada puede llevar a la suposición errónea de que la documentación no es importante. Sin embargo, una documentación adecuada es fundamental para asegurar la continuidad, la calidad y la escalabilidad de los proyectos, así como para facilitar la comunicación y la colaboración dentro del equipo y con los *stakeholders*.

A pesar de esto, a menudo se pasa por alto la documentación en los métodos ágiles debido a su naturaleza iterativa y su ritmo de trabajo rápido. Pero es importante reconocer que su adecuada implementación no solo es importante sino esencial para el éxito a largo plazo de los proyectos ágiles de software. En este contexto, el estándar ISO/IEC/IEEE 26515:2018 *Systems and software engineering — Developing information for users in an agile environment* emergió como una herramienta valiosa que aborda las necesidades específicas de la documentación en entornos ágiles, proporcionando un marco que equilibra la agilidad con la estructura necesaria para la claridad y la continuidad. Por lo tanto, la importancia de la documentación en los métodos ágiles radica en su capacidad para mantener una comunicación clara y efectiva entre los miembros del equipo y los *stakeholders*. Así, una documentación bien mantenida asegura que todos los participantes tengan acceso a la misma información, reduciendo malentendidos y errores que pueden surgir en la ausencia de documentación clara. Además, en equipos con alta rotación de personal, como las organizaciones pequeñas, la documentación es fundamental para que los nuevos miembros se integren rápidamente y comprendan el estado actual del proyecto y las decisiones pasadas.

Por lo tanto, la documentación en los métodos ágiles no debe verse como una tarea secundaria, sino como una parte integral del proceso de desarrollo. Esta no solo facilita la coherencia y la transparencia en el equipo, sino que también garantiza que el conocimiento acumulado durante el proyecto no se pierda, permitiendo una mejor gestión del conocimiento y la toma de decisiones informadas. En última instancia, la combinación de agilidad y una sólida estructura documental proporciona una base sólida sobre la cual se pueden construir proyectos exitosos, adaptables y sostenibles en el tiempo. En este contexto, el estándar ISO/IEC/IEEE 26515:2018 juega un papel crucial al proporcionar directrices específicas que ayudan a los equipos ágiles a crear y mantener

documentación que sea suficiente sin ser excesiva. Este estándar asegura que la documentación sea coherente y estandarizada, lo cual es vital para mantener la calidad del software y facilitar el mantenimiento y escalabilidad futura. La adaptación del estándar a los métodos ágiles permite una integración fluida de la documentación en el ciclo de desarrollo iterativo, promoviendo así la creación de documentos que son útiles y relevantes en lugar de procesos burocráticos.

Sin embargo, la enseñanza de estos estándares del proceso de software presenta desafíos significativos, siendo uno de los principales la resistencia de lo(a)s estudiantes y profesionales a adoptar prácticas de documentación percibidas como anticuadas o burocráticas. Esta percepción se debe en parte a la experiencia previa con sistemas de documentación rígidos y engorrosos que no se adaptan bien a los entornos dinámicos y flexibles que caracterizan a los métodos modernos, como los ágiles. La naturaleza abstracta y técnica de los estándares puede hacer que su enseñanza sea poco atractiva y difícil de entender para lo(a)s estudiantes, quienes a menudo prefieren enfocarse en habilidades prácticas y directamente aplicables en sus carreras profesionales. Además, la falta de ejemplos prácticos y aplicaciones reales pueden dificultar la comprensión y la adopción de estos estándares en la práctica diaria. Es decir, los planes educativos a menudo no incluyen suficientes casos de estudio o escenarios del mundo real donde los estándares se apliquen de manera efectiva, lo que limita la capacidad de lo(a)s estudiantes para comprender la relevancia y los beneficios tangibles de dichos estándares. Esto se agrava por la rápida evolución de la tecnología, que puede hacer que ciertos aspectos de los estándares parezcan desactualizados o irrelevantes. Otra barrera significativa es la falta de formación adecuada de lo(a)s profesore(a)s, quienes pueden no estar completamente familiarizados con los estándares o carecer de los recursos necesarios para enseñarlos de manera efectiva. Sin un entendimiento profundo y una apreciación de la importancia de estos estándares, lo(a)s profesore(a)s pueden no transmitir los conocimientos correctos a lo(a)s estudiantes. También influye la presión constante por acortar los ciclos de desarrollo y entrega de los productos, lo que lleva a la documentación y estandarización a un segundo plano en favor de la velocidad y la agilidad.

En resumen, la enseñanza de estos estándares requiere no solo abordar los aspectos técnicos y abstractos de los mismos, sino también cambiar las percepciones y actitudes hacia la documentación y la estandarización. Pero, además, es necesario dar pasos cruciales para superar estos desafíos, como integrar más ejemplos prácticos a las clases, formar adecuadamente a lo(a)s profesore(a)s y demostrar claramente cómo los estándares contribuyen al éxito a largo plazo de los proyectos. Para abordar estos problemas, los juegos serios se presentan como una alternativa innovadora y eficaz. Los juegos serios son herramientas educativas que combinan elementos de juego con objetivos de aprendizaje, creando una experiencia envolvente y motivadora para los usuarios. En este contexto, un juego serio basado en la dinámica del clásico juego de “Serpientes y escaleras” con una trivia incorporada puede ser una excelente herramienta para complementar la enseñanza y el aprendizaje del estándar ISO/IEC/IEEE 26515:2018. Este juego permite que lo(a)s estudiantes aprendan los aspectos fundamentales del estándar ISO/IEC/IEEE 26515:2018 de una manera interactiva y divertida. Las preguntas de la trivia se relacionan con los principios y directrices del estándar, y el progreso en el juego depende del conocimiento adquirido por lo(a)s estudiantes. De esta manera, considerando la lógica del clásico juego de “Serpientes y escaleras”, si el avatar del(la) estudiante cae en una casilla con la cabeza de una serpiente, éste(a) deberá responder correctamente tres de cinco preguntas para evitar que su avatar descienda y pueda continuar su camino desde ese punto. Al responder incorrectamente, el avatar descenderá por la serpiente y retomará su recorrido desde el nuevo punto inferior. En el caso de caer en una casilla de inicio de una escalera, el(la) estudiante deberá responder correctamente tres de cinco preguntas para subir por la escalera y avanzar rápidamente por el atajo. De lo contrario, el avatar deberá continuar su camino sin hacer uso de la escalera.

El juego propuesto consta de tres niveles de dificultad para adaptarse a los distintos niveles de conocimiento de lo(a)s estudiantes. En el nivel fácil, el(la) estudiante deberá responder preguntas de verdadero o falso; en el nivel medio, escogerá la opción correcta entre tres posibles respuestas; y en el nivel difícil, deberá introducir por teclado la respuesta que considere correcta. Este enfoque “gamificado” no solo incrementa la motivación y el compromiso de lo(a)s estudiantes, sino que también facilita la retención del conocimiento al proporcionar una aplicación práctica de lo aprendido. Además, al incorporar elementos lúdicos y de competencia, se crea un ambiente de aprendizaje más dinámico y efectivo, donde lo(a)s estudiantes pueden reforzar su comprensión del estándar a través de la repetición y la práctica continua.

En resumen, la documentación en los métodos ágiles es esencial para garantizar la comunicación, la calidad y la continuidad de los proyectos de software. El estándar ISO/IEC/IEEE 26515:2018 ofrece un marco valioso de trabajo que adapta la necesidad de documentación a los principios ágiles, fomentando prácticas de documentación eficientes y efectivas. Sin embargo, la enseñanza de estos estándares requiere enfoques innovadores para superar la resistencia y hacer que el aprendizaje sea atractivo. Los juegos serios, como un videojuego de “Serpientes y escaleras” con trivia, representan una solución prometedora para este desafío, combinando entretenimiento y educación para mejorar la comprensión y adopción de estándares en el desarrollo de la documentación ágil. Este enfoque no solo facilita el aprendizaje de los estándares, sino que también promueve una cultura de documentación adecuada y efectiva en el desarrollo ágil de software, asegurando que todos los miembros del equipo comprendan y valoren la importancia de la documentación en el éxito a largo plazo de los proyectos.

## 5.2. Conclusiones derivadas de la evaluación empírica

La evaluación empírica llevada a cabo reveló que el uso del juego serio “*Serpientes y escaleras: Un juego de soporte al aprendizaje del estándar ISO/IEC/IEEE 26515:2018*” es significativamente más efectivo que un enfoque tradicional basado únicamente en clases teóricas y la memorización de conceptos. Como se indicó anteriormente, lo(a)s estudiantes del grupo experimental, que participaron en las sesiones de juego, obtuvieron una calificación promedio del 82% en el examen de conocimientos, en contraste con el 39% logrado por el grupo de control que solamente estudió mediante métodos convencionales. Esta diferencia subraya la capacidad del juego serio para mejorar tanto la comprensión como la retención de los conceptos teóricos. Otro aspecto crucial del juego serio es su impacto positivo en la motivación y el compromiso de lo(a)s estudiantes. La dinámica lúdica y competitiva del juego, combinada con el incentivo de un premio, logró mantener el interés y la atención de lo(a)s participantes; mientras que, en las clases teóricas tradicionales, se observó una disminución de la concentración después de los primeros minutos. Por lo tanto, la interacción activa y la experiencia divertida del juego fomentaron un entorno de aprendizaje más dinámico y agradable. Como consecuencia directa, la evaluación del juego por parte de lo(a)s estudiantes fue en general positiva, destacando la facilidad de uso y la protección contra errores. Aunque se identificaron áreas de mejora en términos de estética y personalización, estas críticas no afectaron significativamente la percepción del juego como herramienta educativa efectiva. La mayoría de lo(a)s estudiantes valoró el juego como una forma útil de aprender los conceptos del estándar, superando la eficacia de los métodos tradicionales.

De manera similar, lo(a)s profesores también reconocieron el valor del juego serio como un recurso complementario a la enseñanza tradicional. Aunque algunos mostraron reservas sobre su integración en los cursos reales debido a la falta de experiencia y ajustes necesarios, la aceptación

general del juego como herramienta educativa alternativa fue alta. Este reconocimiento sugiere que los juegos serios pueden ser una adición valiosa a los métodos de enseñanza existentes.

La información recogida con la evaluación demuestra que la incorporación de juegos serios ofrece múltiples ventajas en el aprendizaje de estándares a nivel universitario. En primer lugar, los juegos serios pueden aumentar la motivación y el compromiso de los estudiantes, haciendo el proceso de aprendizaje más atractivo y entretenido. La experiencia lúdica ayuda pues a lo(a)s estudiantes a interactuar de manera más activa con el material de estudio, facilitando una comprensión más profunda y duradera de los conceptos complejos que son abordados por el estándar. Además, los juegos serios proporcionan realimentación inmediata, lo que permite a lo(a)s estudiantes corregir errores y mejorar continuamente su comprensión del contenido abordado en el juego. Esta realimentación constante facilita un aprendizaje más adaptativo y eficiente. Por último, y no menos importante, los juegos también fomentan el desarrollo de habilidades críticas, como la resolución de problemas y la toma de decisiones, que son esenciales en el campo de la Ingeniería de Software. Sin embargo, es importante recalcar que tanto los resultados y las aseveraciones presentadas en esta tesis no pueden generalizarse, puesto que esto requeriría de una mayor cantidad de evaluaciones en diferentes universidades con una mayor cantidad de estudiantes.

En resumen, el aprendizaje basado en juegos promueve una experiencia más activa y contextualizada, lo que contribuye a un aprendizaje más significativo. Así, al involucrar a lo(a)s estudiantes de manera práctica y experiencial, los juegos serios no solo refuerzan los conceptos teóricos, sino que también los preparan para enfrentar desafíos profesionales futuros con mayor competencia y confianza. De esta manera, la incorporación de los juegos serios en la enseñanza de estándares no solo mejora la motivación y el rendimiento académico, sino que también enriquece la experiencia de aprendizaje al hacerla más interactiva y efectiva. Por estas razones, es altamente recomendable que los estudiantes de licenciatura aprendan sobre estándares a través de juegos serios, aprovechando sus beneficios para desarrollar una comprensión más sólida y duradera de los conceptos clave.

### **5.3. Aspectos para mejorar en el juego serio propuesto**

A pesar de los resultados positivos derivados de la evaluación del juego propuesto para la enseñanza del estándar ISO/IEC/IEEE 26515:2018, durante las entrevistas se identificaron varias áreas que se podrían mejorar para hacer un juego más dinámico y atractivo para lo(a)s estudiantes. A continuación, se presentan algunas sugerencias compartidas por estudiantes y profesores para mejorar la experiencia de juego y maximizar su efectividad como herramienta educativa:

- Incorporación de elementos de gamificación avanzada: Para aumentar la dinámica del juego, se podrían integrar elementos de gamificación adicionales que mantengan a lo(a)s estudiantes más involucrados durante las sesiones de juego, como los siguientes:
  - Sistema de puntos y recompensas: Implementar un sistema de puntos que permita que lo(a)s estudiantes puedan acumular su puntaje a medida que responden correctamente las preguntas. Estos puntos podrían ser canjeados posteriormente por recompensas dentro del juego, como accesorios para la personalización de avatares. Esto incentivaría a lo(a)s estudiantes a esforzarse más y a mantener un interés constante en el juego.
  - Desafíos y logros: Introducir desafíos adicionales o logros que los jugadores puedan alcanzar, tales como completar un nivel sin cometer errores o responder a un número específico de preguntas correctamente. Estos logros podrían ser visualizados en un perfil de usuario, fomentando un sentido de competencia y logro.

- Mejoras en la interfaz y estética del juego: Una interfaz más atractiva para el juego y una mejor personalización pueden contribuir significativamente a mejorar la experiencia del usuario. Esto se podría hacer mediante:
  - Opciones de personalización: Ampliar las opciones de personalización para los avatares y los elementos del juego, con el fin de que lo(a)s estudiantes puedan elegir entre una variedad de aspectos visuales. Esto podría incluir diferentes estilos de avatares, fondos, y temas de tablero independientes de la dificultad elegida, lo que aumentaría la motivación del estudiante al ofrecerle una experiencia visualmente atractiva y personalizada.
  - Interfaz más Intuitiva: Rediseñar la interfaz del juego para que sea más intuitiva y fácil de usar, posiblemente añadiendo tutoriales interactivos al inicio del juego. Esto garantizaría que lo(a)s estudiantes, con poca o nula experiencia con los juegos digitales, puedan comenzar a jugar sin dificultades y entiendan mejor las reglas del juego.
- Integración de retroalimentación inmediata y explicativa: La realimentación inmediata es crucial para el proceso de aprendizaje y podría ser incorporada de manera efectiva a través de:
  - Realimentación detallada: Proporcionar realimentación detallada y explicativa después de cada respuesta, no solo indicando si la respuesta es correcta o incorrecta, sino también explicando el concepto relacionado con la pregunta. Esto contribuiría a que lo(a)s estudiantes entiendan mejor los temas y corrijan errores en el momento.
  - Sugerencias de mejora: Incluir sugerencias y recomendaciones de mejora basadas en el desempeño de los jugadores. Por ejemplo, si un estudiante tiene dificultades con un tema en particular, el juego podría ofrecer recursos adicionales o recomendaciones de estudio.
- Incorporación de elementos narrativos: Los elementos narrativos y cinematográficos pueden hacer que el juego sea más envolvente y atractivo para lo(a)s estudiantes. Esto se podría hacer de la siguiente manera:
  - Historia y contexto: Integrar una historia o narrativa en el juego que se desarrolle a medida que lo(a)s estudiantes avancen. Así, una trama interesante y relevante puede aumentar la inmersión y el disfrute del juego.
  - Cinemáticas y animaciones: Añadir cinemáticas o animaciones para hacer que el juego sea más atractivo visualmente y para proporcionar una narrativa más rica. Estas cinemáticas podrían explicar el contexto o el trasfondo del contenido del juego de manera divertida.
- Mejoras en la interactividad y experiencia del usuario: La interactividad es clave para mantener el interés de lo(a)s estudiantes en el juego. Esto podría lograrse si se agregan los siguientes elementos:
  - Interacción en tiempo real: Incorporar elementos de interacción en tiempo real, como mini juegos o desafíos que se resuelvan en grupo, para hacer el juego más dinámico y participativo.
  - Realimentación del jugador: Permitir que lo(a)s estudiantes proporcionen realimentación sobre el juego para identificar áreas de mejora y adaptarse a las preferencias y necesidades de lo(a)s mismo(a)s.

- Análisis y reportes del desempeño: La capacidad de evaluar y seguir el progreso es esencial para un aprendizaje efectivo y esto se podría lograr si lo(a)s profesore(a)s reciben:
  - Informes de desempeño: Implementar un sistema de informes que permita a lo(a)s profesore(a)s revisar el desempeño y el progreso de lo(a)s estudiantes. Esto ayudaría a identificar áreas de mejora y a adaptar estrategias de enseñanza/aprendizaje personalizadas.
  - Visualización del progreso: Ofrecer gráficos y estadísticas sobre el desempeño de lo(a)s estudiantes en el juego para motivarlos a mejorar y a ver sus logros de manera clara.

Al incorporar estas mejoras en el juego se podría lograr que fuera más dinámico y eficaz en el proceso educativo. Además, estas mejoras no solo pueden aumentar la motivación y el compromiso de lo(a)s estudiantes, sino que también pueden mejorar la comprensión y la retención de los conceptos clave, haciendo que el aprendizaje de los estándares del proceso de software y otros temas complejos sea una experiencia más interactiva y enriquecedora.

## 6. Bibliografía

Albrecht, A. J. (1979). Measuring application development productivity. In Proc. Joint share, Guide, and IBM Application Development Symposium (pp. 83-92).

Anderson, L. W., Krathwohl, D. R. (2001). A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. New York, NY: Longman.

Anicic, K. P., Stapic, Z. (2022). Teaching methods in Software Engineering: Systematic review. *IEEE Software*, 39(6): 73-79.

Arsenault, D. (2009). Video game genre, evolution and innovation. *Eludamos: Journal for computer game culture*, 3(2): 149-176.

Assyne, N., Ghanbari, H., Pulkkinen, M. (2022). The state of research on software engineering competencies: A systematic mapping study. *Journal of Systems and Software*, 185, 111183.

Avdeenko, T., Murtazina, M. (2018). Intelligent support of requirements management in agile environment. In: Borangiu T., Trentesaux D., Thomas A., Cavalieri S. (Eds.), *Studies in Computational Intelligence*, vol. 803. pp. 97-108. Springer, Cham.

Aydan, U., Yilmaz, M., Clarke, P. M., O'Connor, R. V. (2017). Teaching ISO/IEC 12207 software lifecycle processes: A serious game approach. *Computer Standards & Interfaces*, 54: 129-138.

Barrientos-Avendaño, E., Areniz-Arévalo, Y. (2019). "Universidad inteligente: Oportunidades y desafíos desde la Industria 4.0" *Revista Ingenio*, 16(1): 56-60.

Bernhaupt, R. (2011). User experience evaluation in entertainment and games. In: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (Eds.), *Human-Computer Interaction – INTERACT 2011. INTERACT 2011. Lecture Notes in Computer Science*, vol. 6949. Springer, Berlin, Heidelberg.

Bodnar, A., Anastasio, D., Enszer, J. A., Burkey, D. D. (2015). Engineers at play: Games as teaching tools for undergraduate engineering students. *Journal of Engineering Education*, 105(1): 147-200.

Bonilla-Rivas, E., Muñoz, M., Negrón, A. P. P. (2021, October). Strategy for training in the ISO/IEC 29110 standard based on a serious game. In *2021 10<sup>th</sup> International Conference On Software Process Improvement (CIMPS)* (pp. 74-83). IEEE.

Bourgonjon, J., De Grove, F., De Smet, C., Van Looy, J., Soetaert, R., Valcke, M. (2013). Acceptance of game-based learning by secondary school teachers, *Computers & Education*, 67: 21-35.

Calderón A., Ruiz M. (2016). Coverage of ISO/IEC 12207 software lifecycle process by a simulation-based serious game” In: Clarke P., O’Connor R., Rout T., Dorling A. (Eds.), *Communications in Computer and Information Science*, vol 609. pp. 59-70. Springer, Cham.

Calderón, A., Ruiz, M., O’Connor, R. V. (2018). A serious game to support the ISO 21500 standard education in the context of software project management. *Computer Standards & Interfaces*, 60: 80-92.

Calderón, A., Ruiz, M., O’Connor, R. V. (2018). A multivocal literature review on serious games for software process standards education. *Computer Standards & Interfaces*, 57: 36-48.

Calderón, A., Trinidad, M., Ruiz, M., O’Connor, R. V. (2018, October). Teaching software processes and standards: A review of serious games approaches. In *International Conference on Software Process Improvement and Capability Determination* (pp. 154-166). Springer, Cham.

Carnicero, I., González Gaya, C., Rosales Prieto, V. F. (2021, March). Propuestas sobre el uso de la digitalización y metodologías ágiles en la formación en enseñanzas técnicas. *Proc. of the 25<sup>th</sup> International Congress on Project Management and Engineering (CIDIP 2021)* (pp. 2142-2156). AEPRO.

Cattell, R. B., Eber, H. W., Tatsuoka, M. M. (1970). Handbook for the sixteen personality factor questionnaire (16 PF): In clinical, educational, industrial, and research psychology, for use with all forms of the test. Institute for personality and ability testing.

Cico, O., Jaccheri, L., Nguyen-Duc, A., Zhang, H. (2021). Exploring the intersection between software industry and Software Engineering education – A systematic mapping of Software Engineering trends” *Journal of Systems and Software*, 172: 110736.

Coleman, G., O’Connor, R. (2008a). Investigating software process in practice: A grounded theory perspective. *Journal of Systems and Software*, 81(5): 772-784.

Coleman G. O’Connor R. (2008b). The influence of managerial experience and style on software development projects. *International Journal of Technology, Policy and Management*, 8(1): 91-109.

Dauda, I. A., Nuhu, B. K., Abubakar, J., Abdullahi, I. M., Maliki, D. (2021). Software failures: A review of causes and solutions. *Journal of Science Technology and Education*, 9(1): 415-423.

de León-Sigg, M., Villa-Cisneros, J. L., Solís-Recéndez, B. E. (2020). Uso del estándar ISO/IEC 29110 para entrenar estudiantes en procesos de Ingeniería de Software. *Revista Ibérica de Sistemas y Tecnologías de la Información*, (40): 60-72.

de Lope, R. P., Medina-Medina, N., Paderewski, P., Vela, F. L. G. (2015, January). Design methodology for educational games based on interactive screenplays. In *II Congreso de la Sociedad Española para las Ciencias del Videojuego* (pp. 90-101). SECIVI.

Deterding, S., Khaled, R., Nacke, L. E., Dixon, D. (2011, May). Gamification: Toward a definition. In *2011 Annual Conference on Human Factors in Computing Systems (CHI 2011)* (pp. 12-15). ACM Publisher.

Duvall, S., Hutchings, D., Kleckner, M. (2017). “Changing perceptions of discrete mathematics through Scrum-based course management practices” *Journal of Computing Sciences in Colleges*, 33(2): 182-189.

Franch, X., Glinz, M., Mendez, D., Seyff, N. (2021). A study about the knowledge and use of requirements engineering standards in industry. *IEEE Transactions on Software Engineering*, 48(9), 3310-3325.

Galvis, A. (1992). *Ingeniería de Software Educativo*. Madrid, España: Ediciones Uniandes.

- Gamboa, J. Z. (2018). Evolución de las metodologías y modelos utilizados en el desarrollo de software. *INNOVA Research Journal*, 3(10): 20-33.
- García, I., Pacheco, C., Leon, A., Calvo-Manzano, J. A. (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., Felderer, M. (2019). Aligning Software Engineering education with industrial needs: A meta-analysis. *Journal of Systems and Software*, 156, 65-83.
- Goleman, D. (1996). Emotional intelligence. Why it can matter more than IQ. *Learning*, 24(6): 49-50.
- Gómez, S. M. (2020). Aplicación de las metodologías ágiles al proceso de enseñanza-aprendizaje universitario. *Revista d'Innovació Docent Universitària*, (12): 62-73.
- Gorbunova, T. N., Papchenko, E. V., Bazhenov, R. I., Putkina, L. V. (2018). Professional standards in engineering education and Industry 4.0. In 2018 IEEE International Conference on Quality Management, Transport and Information Security, and Information Technologies (IT&QM&IS) (pp. 638-642). IEEE.
- Gotterbarn, D., Miller, K., Rogerson, S. (1999). Computer society and ACM approve software engineering code of ethics. *Computer*, 32(10), 84-88.
- IEEE Computer Society. (2014). Software Engineering Competency Model (SWECOM), Version 1.0. Disponible en: <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). Disponible en: <https://www.acm.org/binaries/content/assets/education/se2014.pdf>
- Insights Team. (2018). 4 ways to ease the transition from waterfall to agile. Forbes. Disponible en: <https://www.forbes.com/sites/insights-scrumalliance/2018/11/01/4-ways-to-ease-the-transition-from-waterfall-to-agile/?sh=72907a93a6fd>
- ISO/IEC (2010). ISO/IEC 26514:2008 Systems and software engineering — Requirements for designers and developers of user documentation. Geneva, Switzerland.
- ISO/IEC/IEEE (2018). ISO/IEC/IEEE 26515:2018 – Systems and software engineering — Developing user documentation in an agile environment. Geneva, Switzerland.
- Jabbari, R., bin Ali, N., Petersen, K., Tanveer, B. (2016, May). What is DevOps? A systematic mapping study on definitions and practices. In *Proceedings of the Scientific Workshop Proceedings of XP2016* (pp. 1-11).
- Kamthan, P., Shahmir, N. (2016, January). Modeling negative user stories is risky business. In *17<sup>th</sup> International Symposium on High Assurance Systems Engineering (HASE)* (pp. 236-237). IEEE.
- Keller, J. M. (2009). *Motivational design for learning and performance: The ARCS model approach*. New Yor, NY: Springer Science & Business Media.
- Konzack, L. (2015). Video game genres. In *Encyclopedia of Information Science and Technology*, Third Edition (pp. 3070-3076). IGI Global.
- Kosa, M., Yilmaz, M., O'Connor, R., Clarke, P. (2016). Software engineering education and games: A systematic literature review. *Journal of Universal Computer Science*, 22(12): 1558-1574.
- Kühn, F. D. (2019). Proceso de implementación de analíticas de aprendizaje en juegos serios dentro de espacios educativos. *e-tramas*, (3): 20-39.

Kuz, A., Falco, M., Giandini, R. S. (2018). Comprendiendo la aplicabilidad de Scrum en el aula: Herramientas y ejemplos. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, (21): 62-70.

Laporte, C. Y., April, A. (2018). *Software engineering standards and models*. Wiley-IEEE Computer Society Press.

Laporte, C. Y., Muñoz, M., Mejia Miranda, J., O'Connor, R. V. (2018). Applying software engineering standards in very small entities: From startups to grownups. *IEEE Software*, 35(1), 99-103.

Laporte, C. Y., Muñoz, M. (2021). Not teaching software engineering standards to future software engineers-malpractice? *Computer*, 54(5): 81-88.

León, A. (2016). *Uso del aprendizaje basado en juegos para el diseño de una herramienta de soporte a la enseñanza de la elicitación de requisitos de software en un contexto universitario*. (Tesis de Licenciatura. NovaUniversitas).

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, 22(140): 55-63.

López, J. L. O. (2018). Scrum como estrategia para el aprendizaje colaborativo a través de proyectos. Propuesta didáctica para su implementación en el aula universitaria. Profesorado. *Revista de Currículum y Formación del Profesorado*, 22(2): 509-527.

Manan, A., Suryaman, E. (2021). Improving student competency in Software Engineering through training. *The Journal of Worker Competence and Performance (JWCP)*, 1(01): 1-9.

Marín, B. (2022). Lessons learned about gamification in Software Engineering education. In Khosrow-Pour, M. (Ed.), *Research Anthology on Developments in Gamification and Game-Based Learning*, vol. 4. pp. 1473-1496. IGI Global.

Masuda, S., Nishi, Y., Suzuki, K. (2020, October). Complex software testing analysis using international standards. In *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 241-246). IEEE.

Mathrani, A., Wickramasinghe, S., Jayamaha, N. P. (2021). An evaluation of documentation requirements for ISO 9001 compliance in Scrum projects. *The TQM Journal*, 34(5): 901-921.

Matturro, G., Raschetti, F., Fontán, C. (2019). A systematic mapping study on soft skills in Software Engineering. *Journal of Universal Computer Science*, 25(1): 16-41.

Meding, W., Staron, M. (2020, April). Making software measurement standards understandable. In *24<sup>th</sup> International Conference on Evaluation and Assessment in Software Engineering* (pp. 368-370). ACM Publisher.

Moder, J. J., Phillips, C. R., Davis, E. W. (1983). *Project Management with CPM, PERT and Precedence Diagramming*, 3<sup>rd</sup> Edition, New York, NY: Van Nostrand Reinhold.

Moody, D. L., Sindre, G. (2003). Evaluating the effectiveness of learning interventions: An information systems case study, In *11<sup>th</sup> European Conference on Information Systems (ECIS 2003)* (pp. 1311-1326). ACPI.

Montero, B. M., Cevallos, H. V., Cuesta, J. D. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales Revista Multidisciplinaria de Investigación*, 2(17): 114-121.

Moura, V., Santos, G. (2018, March). ProcSoft. In *17<sup>th</sup> Brazilian Symposium on Software Quality* (pp. 122-134).SBQS.

- Muñoz, M., Mejia, J., Laporte, C. Y. (2018). Implementación del estándar ISO/IEC 29110 en centros de desarrollo de software de universidades mexicanas: Experiencia del Estado de Zacatecas. *RISTI-Revista Ibérica de Sistemas y Tecnologías de la Información*, (29): 43-54.
- Muñoz, M., Mejia, J., Peña, A., Lara, G., Laporte, C. Y. (2019). Transitioning international software engineering standards to academia: Analyzing the results of the adoption of ISO/IEC 29110 in four Mexican universities. *Computer Standards & interfaces*, 66: 103340.
- Nakagawa, E. Y., Antonino, P. O., Schnicke, F., Kuhn, T., Liggesmeyer, P. (2021). Continuous systems and Software Engineering for industry 4.0: A disruptive view. *Information and Software Technology*, 135: 106562.
- O'Connor, R. V. (2019). Software development process standards for very small companies. In Mehdi Khosrow-Pour, D.B.A. (Ed.), *Advanced Methodologies and Technologies in Digital Marketing and Entrepreneurship* (pp. 681-694). IGI Global.
- Oguz, D., Oguz, K. (2019). Perspectives on the gap between the software industry and the Software Engineering education. *IEEE Access*, 7: 117527-117543.
- Ortiz-Ramírez, J. M., Bravo-Agapito, J. (2019). Gamificación aplicada a la educación: Videojuego Serpientes y Escaleras. *Revista Tecnología, Ciencia y Educación*, 149-166.
- Parnell, M. J., Berthouze, N., Brumby, D. (2009). Playing with scales: Creating a measurement scale to assess the experience of video games. London, UK: University College London.
- Petri, G., von Wangenheim, C. G., Borgatto, A. F. (2019). MEEGA+: Um modelo para a avaliação de jogos educacionais para o ensino de computação. *Revista Brasileira de Informática na Educação*, 27(03): 52-81.
- Pieper, J., Lueth, O., Goedicke, M., Forbrig, P. (2017, April). A case study of Software Engineering methods education supported by digital game-based learning: Applying the SEMAT essence kernel in games and course projects. In *2017 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1689-1699). IEEE.
- Plass, J. L., Homer, B. D., Kinzer, C. K. (2015). Foundations of game-based learning. *Educational Psychologist*, 50(4): 258-283.
- Ramírez, A., Aiello, A., Lincke, S. J. (2020, November). A survey and comparison of secure software development standards. In *13<sup>th</sup> CMI Conference on Cybersecurity and Privacy (CMI)-Digital Transformation-Potentials and Challenges* (pp. 1-6). IEEE.
- Rodríguez, G. H., Borromeo, N., Gonzalez, P. (2019). Un enfoque de serious gaming para la enseñanza de Scrum. In *I Simposio Argentino de Educación en Informática (SAEI 2019)* (pp. 245-253). SADIO.
- Rodríguez, G., Glessi, M., Teyseyre, A. R. (2020). Una revisión sistemática de juegos serios para la enseñanza de métodos ágiles. *II Simposio Argentino de Educación en Informática (SAEI 2020)* (pp. 453-462). SADIO.
- Rosdahl, J. (2018). Computer society standards drive industry. Qualcomm Technologies, Inc.
- Salza P., Musmarra P., Ferrucci F. (2019). Agile methodologies in education: A review. In: Parsons D., MacCallum K. (Eds.), *Agile and Lean Concepts for Teaching and Learning*. pp. 25-45. Springer, Singapore.
- Sandí-Delgado, J. C., Sanz, C. V., Lovos, E. N. (2022). Acceptance of serious games to develop digital competencies in higher education. *The Electronic Journal of e-Learning*, 20(3): 351-367.

Savi, R., von Wangenheim, C. G., Borgatto, A. F. (2011). A model for the evaluation of educational games for teaching Software Engineering. *25<sup>th</sup> Brazilian Symposium on Software Engineering (SBES 2011)* (pp. 194-203). IEEE.

Sleeter, C., Carmona, J. F. (2017). *Un-standardizing curriculum: Multicultural teaching in the standards based classroom*. Boston, MA: Teachers College Press.

Stephens, R. (2015). *Beginning Software Engineering*. Hoboken, NJ: Wiley & Sons.

Stol, K. J., Schaarschmidt, M., Goldblit, S. (2022). Gamification in Software Engineering: The mediating role of developer engagement and job satisfaction. *Empirical Software Engineering*, 27(2): 1-34.

Torres, L. I. (2020). Estudio comparativo entre metodologías tradicionales y metodologías ágiles aplicadas a proyectos IT en entorno industrial. Trabajo final de Máster. Máster Universitario en Dirección de Proyectos. Universidad Pública de Navarra. Pamplona, España.

Valencia, D. C., Duque, F. A. B. (2023). Technology Acceptance Model (TAM): A study of teachers' perception of the use of serious games in the higher education. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 18(1): 123-129.

Winters, J. (2022). How do we get there from here? Accomplishing Internationalization at a Higher Education Institution (Doctoral dissertation, Northeastern University).

Yasin, A., Fatima, R., Liu, L., Wei, Z., Jianmin, W. (2021, November). Experience report on improving Software Engineering education. In *2021 International Conference on Innovative Computing (ICIC)* (pp. 1-12). IEEE.

## 7. Anexo A.- Acrónimos

3D	Tridimensional
ACM	Asociación de Maquinaria Computacional
AWS	Servicios Web de Amazon
DA	De acuerdo
DEVOPS	Desarrollo y Operaciones
DS	Desviación estándar
ED	En desacuerdo
GBL	Aprendizaje basado en juegos
GDD	Documento de diseño del juego
GDS	Desarrollo global de software
IBM	Corporación Internacional de Máquinas de Negocios
IEC	Comisión Electrotécnica Internacional
IEEE	Instituto de Ingenieros Eléctricos y Electrónicos
IEEE-SA	Asociación de Estándares del IEEE
ISO	Organización Internacional de Normalización
JTC	Comité Técnico Conjunto
M	Media
MDA	Mecánica, Dinámica, Estética
MEEGA	Modelo para la Evaluación de Juegos Educativos para enseñar la Ingeniería de Software
MEEGA+	Modelo para la Evaluación de Juegos Educativos enfocados en la Enseñanza de la Computación
N	Neutral
NPC	Personaje no jugable
SE	Ingeniería de Software
SEEK	Cuerpo de Conocimiento para la Educación de la Ingeniería de Software
SPSS	Paquete Estadístico para las Ciencias Sociales
SUNEO	Sistema de Universidades Estatales de Oaxaca

TA	Totalmente de acuerdo
TI	Tecnologías de la Información
TD	Totalmente en desacuerdo
UNISTMO	Universidad del Istmo
UNPA	Universidad del Papaloapan
UMAR	Universidad del Mar
UNSI	Universidad de la Sierra Sur
UNSIJ	Universidad de la Sierra Juárez
UNCA	Universidad de la Cañada
UTM	Universidad Tecnológica de la Mixteca
SE2014	Lineamientos Curriculares para Programas de Licenciatura en Ingeniería de Software 2014
SWECOM	Modelo de Competencias de la Ingeniería de Software
XP	Programación Extrema

## **8. Anexo B.- Preguntas incluidas en el juego serio**

Como se explicó en el Capítulo 3 de esta tesis, el juego serio llamado “*Serpientes y escaleras: Un juego de soporte al aprendizaje del estándar ISO/IEC/IEEE 26515:2018*” incluye una trivia que ayuda a que los estudiantes repasen de manera amena y divertida los conceptos fundamentales que se relacionan con la creación de información en el contexto de los métodos ágiles, como lo indica el estándar ISO/IEC/IEEE 26515:2018. En este sentido, se presentan a continuación la organización que se siguió para diseñar las preguntas, por cada nivel de complejidad, que correspondieran con las secciones del estándar.

### **8.1. Estructura de las preguntas**

Con el objetivo de diseñar una trivia sobre el estándar ISO/IEC/IEEE 26515:2018 se siguió la estructura lógica que se presenta en el documento oficial del mismo, la cual se basa en las siguientes cuatro secciones:

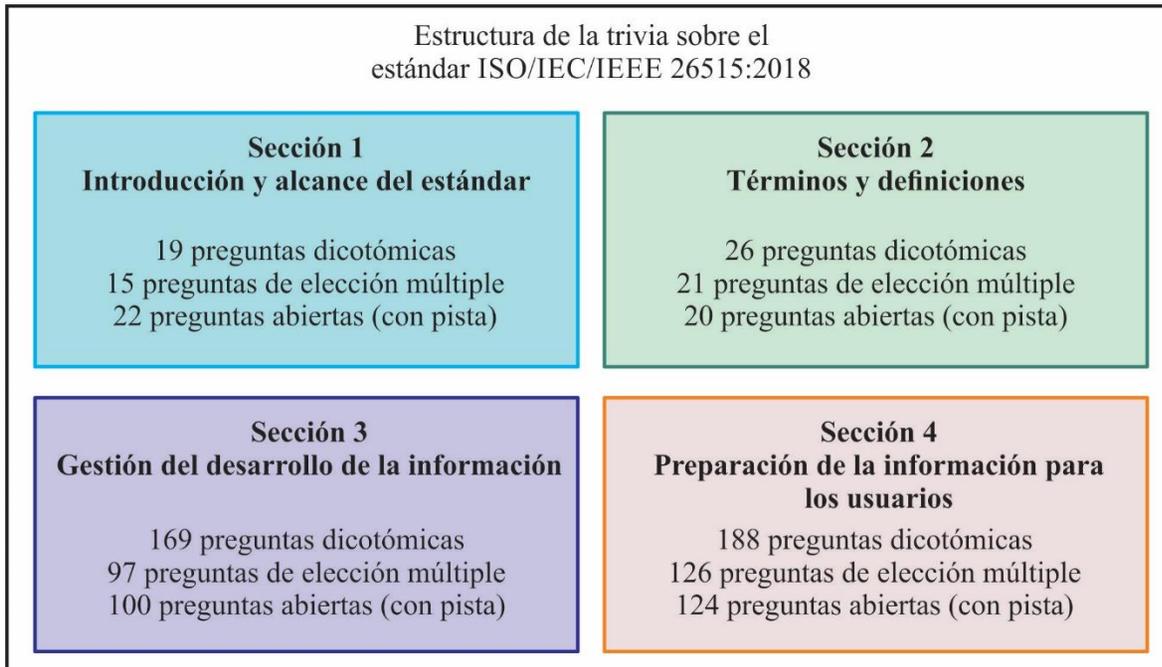
- **Introducción y alcance del estándar:** Que contiene información para ubicar el contexto donde se puede utilizar el estándar y quiénes son los posibles usuarios.
- **Términos y definiciones:** Que contiene un glosario de términos para que el estándar sea mejor comprendido por la audiencia objetivo.
- **Gestión del desarrollo de la información:** Que contiene información sobre el primer proceso fundamental del estándar, el cual establece las actividades necesarias para fundamentar la importancia de que el líder del desarrollo de la información o el líder del proyecto planifiquen un proyecto de desarrollo de software y gestionen las actividades del desarrollo de la información en un entorno ágil.
- **Preparación de la información para los usuarios:** Que contiene información sobre el segundo proceso fundamental del estándar, el cual aborda las actividades necesarias para establecer los requisitos para diseñar, desarrollar y proporcionar información a los usuarios en un entorno ágil.

Considerando lo anterior, se decidió analizar y dividir los contenidos del estándar de tal manera que fuera posible generar tres niveles de complejidad en función del tipo de pregunta, como se menciona en el Capítulo 3 de la tesis. De acuerdo con el nivel de dominio que los estudiantes vayan adquiriendo con el juego, estos tres niveles pueden ser alternados para evaluar el entendimiento de los conceptos. Los tres niveles implementados en la trivia son los siguientes:

- **Nivel fácil:** Preguntas dicotómicas con respuestas de verdadero o falso.

- Nivel medio: Preguntas de opción múltiple.
- Nivel difícil: Preguntas abiertas con una pista sobre el tamaño de la palabra respuesta.

De esta manera, la trivia diseñada para el juego serio “*Serpientes y escaleras: Un juego de soporte al aprendizaje del estándar ISO/IEC/IEEE 26515:2018*” sigue la estructura mostrada en la Figura 8.1.



**Figura 8.1.** Secciones del estándar y cantidad de preguntas por nivel de complejidad

La trivia completa no se agregó al documento de tesis pero puede consultarse en: [https://docs.google.com/document/d/1HS1y\\_YltsaNo734Zy8l0xcDbodlZwO6Z/edit?usp=drive\\_link&ouid=113135657344358622432&rtpof=true&sd=true](https://docs.google.com/document/d/1HS1y_YltsaNo734Zy8l0xcDbodlZwO6Z/edit?usp=drive_link&ouid=113135657344358622432&rtpof=true&sd=true). En este documento se muestra la totalidad de las preguntas por sección y nivel de complejidad y, además, se indican las respuestas correctas en dado caso de que algún profesor(a) quiera utilizarlas para su curso.