

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



**Desarrollo de un simulador de excavación arqueológica para
educación superior**

Tesis para obtener el título profesional de Ingeniero Informático

Autor:

Alejandro Francisco Roman Nureña

Asesores:

Mg. Claudia Maria del Pilar Zapata del Rio

Mg. Luis David Robles Pizarro

Lima, Abril, 2023

Informe de Similitud

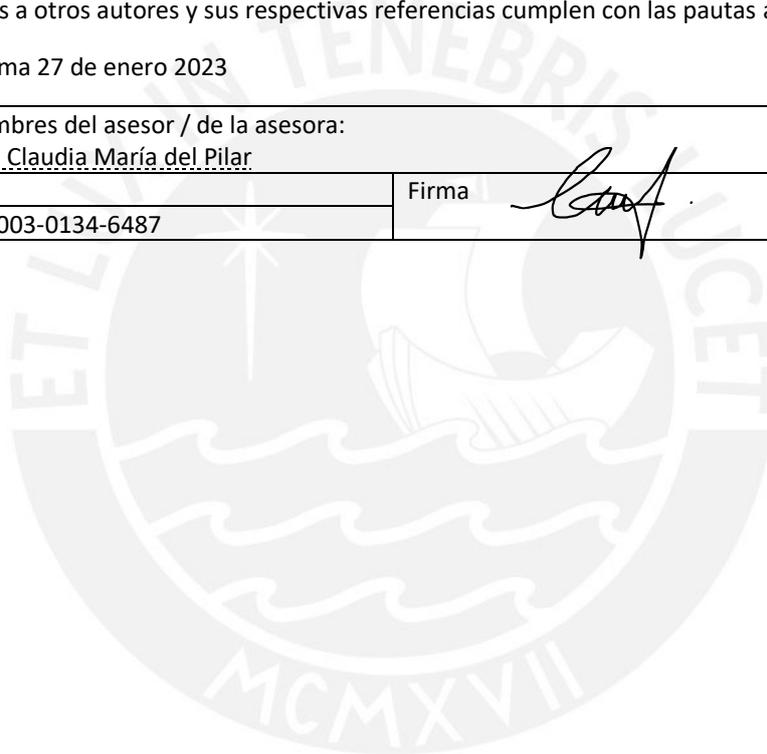
Yo, Claudia María del Pilar Zapata Del Río, docente de la Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú, asesor(a) de la tesis/el trabajo de investigación titulado Desarrollo de un simulador de excavación arqueológica para educación superior, del/de la autor(a)/ de los(as) autores(as) Alejandro Francisco Román Nureña,

dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 13%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 26/01/2023.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: Lima 27 de enero 2023

Apellidos y nombres del asesor / de la asesora: Zapata Del Río, Claudia María del Pilar	
DNI: 10799864	Firma 
ORCID: 0000-0003-0134-6487	



Resumen

La arqueología se encarga del estudio del ser humano a través de los restos materiales que el hombre ha dejado a lo largo de la historia. Debido a que es principalmente a través del proceso de excavación arqueológica que los arqueólogos obtienen estos restos materiales, la formación de futuros arqueólogos hace énfasis en la enseñanza de conocimiento práctico.

Sin embargo, es difícil para las instituciones educativas realizar excavaciones arqueológicas. Esto se debe a que realizarlas es un proceso caro, lento y destructivo. Además, este tipo de actividades se encuentran limitadas por la disponibilidad de los sitios arqueológicos y su cercanía a los centros de estudio.

Por estos motivos, múltiples instituciones educativas han optado por realizar actividades dentro de las aulas que simulen la experiencia de campo. Este fue el caso del curso Arqueología, en el ciclo 2012-1, de la Pontificia Universidad Católica del Perú. Sin embargo, como reconoció el docente del curso, estas actividades no cubren todos los aspectos de una excavación que son de interés académico que se enseñen.

Para incluir los aspectos faltantes, el Grupo Avatar PUCP, junto con alumnos de ingeniería informática, desarrollaron un simulador educativo virtual sobre el proceso de excavación arqueológica. Sin embargo, a pesar de que se obtuvo una simulación más fidedigna, el simulador tuvo problemas de rendimiento debido a su diseño de arquitectura. Esto dificultaba su uso, pues el alto consumo de recursos computacionales provocaba que en ocasiones el simulador se detenga abruptamente durante su ejecución.

Por ello, el presente trabajo propone el desarrollo de un nuevo simulador de excavaciones arqueológicas. Este simulador educativo incluirá los aspectos de la excavación arqueológica que no fueron simulados por las actividades en aula. Así mismo, para permitir una simulación estable, el nuevo simulador tendrá un diseño de arquitectura que permita un consumo eficiente de recursos computacionales. Finalmente, también se considerará la usabilidad del software, ya que este tiene una finalidad educativa, por lo que es importante que pueda ser utilizado por estudiantes de educación superior.

Antes de desarrollar el simulador, el autor recopiló los aspectos del proceso de excavación arqueológica que debería tener el software. Posteriormente, confirmó el listado de aspectos a incluirse con una experta en la docencia de arqueología. De esta manera, se concluyó que el nuevo simulador integra los aspectos del proceso de excavación que son de interés académico que se enseñen.

Posteriormente, tras implementar el simulador, el autor realizó una prueba de estrés comparativa entre el simulador desarrollado y el anterior. A partir de esta comparación, se pudo concluir que el software desarrollado tuvo un menor consumo de recursos computacionales a lo largo de la simulación.

Finalmente, tras llevar a cabo múltiples pruebas con usuarios a lo largo del desarrollo del simulador, también se pudo concluir que el nuevo simulador educativo puede ser utilizado y comprendido por estudiantes de educación superior.

Dedicatoria

Dedico este proyecto de tesis a mi madre, por darme las tres cosas más importantes que se le puede dar a un hijo: Vida, amor y educación.



Tabla de Contenido

Tabla de Contenido	4
Capítulo 1. Generalidades	9
1.1 Problemática	9
1.2 Objetivos	12
1.2.1 Objetivo general	12
1.2.2 Objetivos específicos	12
1.2.3 Resultados esperados	12
1.2.4 Mapeo de objetivos, resultados y verificación	13
1.3 Herramientas y métodos	15
1.3.1 Mapeo de resultados esperados y herramientas y métodos seleccionados	15
1.3.2 Integración de metodología <i>Extreme Programming</i> y <i>User-centered design</i>	17
1.3.2.1 User-centered design	18
1.3.2.2 Extreme Programming	19
1.3.2.3 Metodología a utilizarse	20
1.3.3 Software Architecture Analysis Method	21
1.3.4 Modelo de vista de arquitectura de software "4+1"	23
1.3.5 Lenguaje de programación C++	24
1.3.6 Prueba de estrés y comparación de consumo promedio de recursos computacionales	25
1.4 Alcance, limitaciones y riesgos	25
1.4.1 Alcance	25
1.4.2 Limitaciones	26
1.4.3 Riesgos	27
1.1 Viabilidad	28
	4

1.1.1	Viabilidad técnica	28
1.1.2	Viabilidad temporal	29
1.1.3	Viabilidad económica	29
Capítulo 2. Marco conceptual		31
2.1	Arqueología	31
2.2	Restos materiales	31
2.3	Simulación	31
2.4	Usabilidad	33
2.5	Excavación arqueológica	33
Capítulo 3. Estado del arte		34
3.1	Protocolo de revisión	34
3.2	Revisión y discusión	37
3.3	Conclusiones	39
Capítulo 4. Identificación de los aspectos sobre la excavación arqueológica que puedan aplicar en un simulador		41
4.1	Formulación de historias de usuario con los aspectos del proceso de excavación arqueológica	41
i.	Preparación de la entrevista con experto	41
ii.	Desarrollo de la entrevista con experto	41
iii.	Identificación de personas y escenarios	41
iv.	Formulación del catálogo de requisitos e historias de usuario	41
4.1.1	Preparación de la entrevista	41
4.1.2	Desarrollo de la entrevista	42
4.1.3	Identificación de personas y escenarios	43
4.1.4	Formulación de historias de usuario	45
4.2	Validación académica de los aspectos a ser incluidos en el simulador de excavaciones arqueológicas	45

Capítulo 5. Diseño de la arquitectura de software que soporte la simulación de excavaciones arqueológicas	54
5.1 Diseño de la arquitectura de software que soporte la simulación	54
5.1.1 Vista lógica: Diagrama de clases	55
5.1.2 Vista física: Diagrama de despliegue	59
5.1.3 Vista de proceso: Diagrama de actividades	59
5.1.4 Vista de desarrollo: Diagrama de componentes	61
5.2 Validación del rendimiento de la arquitectura planteada a través del método “Software Architecture Analysis Method”	63
5.3 Prueba de estrés comparativa con el simulador “El Pez Dorado” original	66
5.3.1 Determinación de la normalidad de los conjuntos de datos	68
5.3.2 Definición y comprobación de hipótesis sobre las características de rendimiento	70
Capítulo 6. Desarrollo de un simulador de excavaciones arqueológicas en base a los aspectos identificados	75
6.1 Desarrollo del módulo de simulación del proceso de excavación arqueológica	75
6.2 Desarrollo del módulo de interfaz de usuario del simulador de excavaciones arqueológicas	77
Capítulo 7. Validar la usabilidad del simulador desarrollado con una muestra de estudiantes	79
7.1 Resultado de pruebas de usabilidad de la interfaz de usuario del simulador desarrollado.	79
Capítulo 8. Conclusiones y trabajos futuros	83
8.1 Conclusiones	83
8.2 Trabajos futuros	86
Capítulo 9. Referencias	88

Índice de Figuras

Figura 1. Proceso XP integrado con técnicas UCD. Adaptado de (Aguilar & Zapata, 2017)	21
Figura 2. Vista de la cabina de piloto del simulador. Imagen tomada de (Seven Summits Studio, 2015)	32
Figura 3. Prototipo en papel de pantalla de menú principal (Elaboración propia)	46
Figura 4. Prototipo en papel de pantalla de prospección arqueológica (Elaboración propia)	46
Figura 5. Prototipo en papel de pantalla de excavación arqueológica (Elaboración propia).	47
Figura 6. Prototipo de pantalla de menú principal (Elaboración propia)	48
Figura 7. Prototipo de pantalla de detalle de método de muestreo (Elaboración propia)	49
Figura 8. Prototipo de la pantalla de gestión de recursos (Elaboración propia)	50
Figura 9. Prototipo de la pantalla de excavación (Elaboración propia)	51
Figura 10. Prototipo de inventario de artefactos encontrados (Elaboración propia)	52
Figura 11. Clase "Game" (Elaboración propia)	55
Figura 12. Clase "ResourceManager" (Elaboración propia)	56
Figura 13. Clase "Scene" (Elaboración propia).	57
Figura 14. Clase "GameState" (Elaboración propia).	58
Figura 15. Vista física: Diagrama de despliegue (Elaboración propia)	59
Figura 16. Vista de proceso: Diagrama de actividades - Proceso de menú principal (Elaboración propia)	60
Figura 17. Vista de desarrollo: Diagrama de componentes	62
Figura 18. Resultados de prueba Shapiro-Wilk sobre las mediciones del simulador antiguo y el simulador nuevo (Elaboración propia).	69
Figura 19. Resultados de prueba U-Mann Whitney sobre las mediciones del simulador antiguo y el simulador nuevo (Elaboración propia).	71

Figura 20. Comparación de porcentaje de uso de tiempo de CPU promedio por configuración (Elaboración propia).	72
Figura 21. Comparación de consumo de memoria virtual promedio por comparación (Elaboración propia).	73
Figura 22. Comparación de cantidad de hilos promedio utilizados por configuración (Elaboración propia).	74

Índice de Tablas

Tabla 1. Relación entre objetivos específicos, resultados esperados y medios de verificación (Elaboración propia).	13
Tabla 2. Mapeo de resultados esperados y herramientas y métodos seleccionados (Elaboración propia)	15
Tabla 3. Doce prácticas principales de Extreme Programming. Adaptado de (Akpata & Riha, 2004).	19
Tabla 4. Mapeo de vistas y diagramas utilizados (Elaboración propia).	23
Tabla 5. Mapeo de riesgos, impacto estimado y medidas adoptadas (Elaboración propia).	27
Tabla 6. Razones de viabilidad técnica por objetivo específico (Elaboración propia).	28
Tabla 7. Formulación de preguntas de investigación a partir de método PICO (Elaboración propia).	35
Tabla 8. Criterios de inclusión y exclusión utilizados (Elaboración propia).	36
Tabla 9. Cantidad de documentos por clasificación (Elaboración propia).	37
Tabla 10. Descripción del escenario 1 (Elaboración propia)	43
Tabla 11. Descripción del escenario 2 (Elaboración propia)	44
Tabla 12. Actividades 2 y 3 del método SAAM (Elaboración propia)	64
Tabla 13. Parámetros de prueba de rendimiento	67

Capítulo 1. Generalidades

1.1 Problemática

La arqueología se puede definir como el estudio del pasado del ser humano a través de restos materiales, es decir, los objetos utilizados, modificados o creados por el hombre a lo largo de la historia (Archaeological Institute of America, 2007). A diferencia del estudio de la historia, el objeto de estudio de la arqueología no se basa en restos escritos, sino en restos materiales, como ruinas, herramientas, cerámicos, ropa, tumbas, huesos, semillas, entre otros (Barker, Roberts, Gosden, Horning, & Welham, 2016, p. 5). La importancia de este campo de estudio reside en la interpretación de estos restos materiales, ya que permite entender el pasado y evolución tanto del hombre como de las sociedades humanas (Quality Assurance Agency for Higher Education, 2014, p. 6).

Debido a la diversidad de restos materiales que estudia, la arqueología inevitablemente hace uso de teorías y técnicas analíticas de otros campos de estudio, como los de humanidades, ciencias sociales y ciencias naturales (Barker et al., 2016, p. 5). De esta manera, se comprende que la enseñanza de futuros arqueólogos se brinde con la enseñanza de conocimientos teóricos de estos campos de estudio, para hacer uso de sus teorías y prácticas (Barker et al., 2016, p. 17).

Por otra parte, los programas de educación de arqueología incluyen también la enseñanza de conocimiento práctico (Quality Assurance Agency for Higher Education, 2014, p. 16). Esto es necesario debido a que el trabajo de campo es el medio principal que tienen los arqueólogos para recuperar evidencias arqueológicas (Quality Assurance Agency for Higher Education, 2014, p. 16). De esta manera, se entiende que la enseñanza de conocimiento práctico es un aspecto esencial en los programas educativos de arqueología (Quality Assurance Agency for Higher Education, 2014, p. 16). Sin embargo, a pesar de que la mayoría de estudiantes, al ser encuestados, afirman que las experiencias reales en campo les permite reforzar los conocimientos adquiridos (Croucher, Cobb, & Brennan, 2008, p. 29), es difícil para las instituciones proveer este tipo de enseñanza (Croucher et al., 2008, p. 44). Esto se debe a que realizar una excavación arqueológica real, entendida como el proceso en el cual se excavan los suelos para recuperar restos materiales, es un proceso destructivo, caro y lento (Reilly, 1991, p. 133). Además, este tipo de enseñanza se encuentra limitada por la disponibilidad de los sitios arqueológicos y la cercanía estos a los centros de estudio (Getchell, Miller, Nicoll, Sweetman, & Allison, 2010, p. 283). Por estos motivos, múltiples

instituciones educativas han brindado actividades que permitan aprender y experimentar sobre aspectos de la arqueología sin tener que salir de las aulas de clase (Archaeology Education Clearinghouse, n.d.; Smithsonian National Museum of Natural History, n.d.; Society for American Archaeology, n.d.-a). Como ejemplos de estas actividades se encuentra la interpretación de fotografías de sitios arqueológicos (Kauppa, 2009), de residuos materiales de una casa (Brown, 2011) o de los restos de un cerámico (Ashmore & Baumann, 2003).

En el Perú, se impartió en la Pontificia Universidad Católica del Perú (PUCP) el curso *Arqueología* (ARQ107) durante el semestre 2012-1¹. Acorde a la estructura de la enseñanza de la arqueología anteriormente descrita, este curso brindó un enfoque teórico y práctico en el estudio de la arqueología, optando por el uso de módulos de actividades en el segundo enfoque². Entre estas actividades, se encuentra la simulación, es decir, la imitación del comportamiento, del proceso de excavación a través de la representación de un sitio arqueológico en hojas de papel (Rucabado, 2010, p. 2). A pesar de que el uso de esta actividad presentó beneficios respecto a las excursiones a sitios arqueológicos, como el ahorro de la inversión necesaria para realizar una excursión, también se reconoció que presenta un problema respecto a la fidelidad de la simulación. (Rucabado, 2010, p. 4). Esto se debe a que esta actividad en clase no consideraba múltiples aspectos del proceso de excavación arqueológica que afectan en la vida real al proceso y que es de interés académico que se enseñen sobre la excavación arqueológica (Rucabado, 2010, p. 4). Algunos aspectos no considerados en esta actividad que sí se incluirán en el presente proyecto son:

- Manejo de recursos del proyecto, como dinero, tiempo y personal (Rucabado, 2010, p. 4)
- Posibilidad de destrucción de restos materiales debido a un incorrecto manejo de las herramientas de excavación (Rucabado, 2010, p. 4)
- Método de registro de los hallazgos (Rucabado, 2010, p. 4)

Debido a este problema de la actividad en clase utilizada (Rucabado, 2010, p. 5), se optó por enseñar el proceso de excavación arqueológica a través del uso de un

^{1 2} "Arqueología – Programa Analítico 2012-1" - PUCP

simulador educativo virtual en reemplazo de la actividad en clase descrita (Baldeón, 2010, p. 3). De esta manera, con el apoyo del Grupo Avatar y los alumnos del curso de Lenguajes de Programación 2, se desarrolló un simulador educativo virtual sobre el proceso de excavación arqueológica (Baldeón, 2010, p. 3) que sí incluía entre sus funcionalidades los aspectos identificados que no fueron considerados en la actividad en clase mencionada (Rucabado, 2010, p. 5). Sin embargo, a pesar de haber resuelto el problema de la fidelidad de la simulación, el simulador virtual desarrollado introdujo consigo un nuevo problema a la simulación: el rendimiento del software desarrollado (L. D. Robles, comunicación personal, mayo 25, 2018)³. De acuerdo al magíster Luis Robles, encargado de la revisión y mejora técnica del simulador, este problema se hacía evidente cuando, al hacer uso del simulador, este detenía su funcionamiento de manera abrupta o provocaba que la computadora en la cual se utilizaba el simulador se calentara por el esfuerzo que realizaba para ejecutar el simulador (L. D. Robles, comunicación personal, mayo 25, 2018). Posteriormente, se realizó una revisión del simulador y se determinó que este problema de rendimiento se debía a que el simulador utilizaba muchos recursos computacionales de la computadora que lo ejecutaba (L. D. Robles, comunicación personal, mayo 25, 2018). Este mayor consumo de recursos computacionales, a su vez, ocasionaba que el simulador se detuviera intempestivamente. Asimismo, con la revisión del simulador se determinó también que este mayor consumo de recursos se debía a un incorrecto diseño de arquitectura de software utilizado (L. D. Robles, comunicación personal, mayo 25, 2018).

En resumen, se ha expuesto sobre la necesidad de hacer uso de actividades de clase que simulen el proceso de excavación arqueológica con fines académicos. Asimismo, también se ha expuesto sobre las simulaciones realizadas, tanto con actividades en aulas como con simuladores virtuales, que se han utilizado en la PUCP para simular este proceso. Sin embargo, se han identificado dos problemas principales para llevar a cabo la simulación del proceso de excavación arqueológica. El primero de ellos es que no se incluyen múltiples aspectos de la excavación arqueológica que sí afectan al proceso de excavación en la vida real, como el manejo de recursos. El segundo problema identificado es el inadecuado rendimiento del simulador desarrollado, debido a un mal diseño de arquitectura de software sobre el cual se basó el simulador.

Ante la problemática descrita, se propone el desarrollo de un simulador de excavación

³ La transcripción de la entrevista se proporciona en el Anexo B.

arqueológica para educación superior que sí considere tanto los aspectos relacionados al proceso de excavación, como una arquitectura de software que permita una simulación estable. Además, se ha considerado conveniente también proponer una validación de usabilidad del simulador a desarrollarse. Esta usabilidad a validarse debe entenderse como el grado en el cual los usuarios puedan hacer uso correctamente del simulador. Esta decisión se debe a que, ya que el simulador tiene una finalidad educativa, se requiere asegurarse que los estudiantes de educación superior puedan hacer uso correcto del simulador y no presenten inconvenientes durante su manipulación.

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar un simulador de excavaciones arqueológicas para educación superior, que considere la enseñanza de los diferentes aspectos de la excavación arqueológica como el uso de recursos, usando una arquitectura de software que permita su funcionamiento eficiente y con un diseño que sea usable para los estudiantes.

1.2.2 Objetivos específicos

Los objetivos específicos del presente proyecto son:

- O1. Identificar aspectos sobre la excavación arqueológica que se puedan aplicar en un simulador
- O2. Diseñar la arquitectura de software que soporte la simulación de excavaciones arqueológicas usando de modo eficiente los recursos del sistema
- O3. Desarrollar un simulador de excavaciones arqueológicas en base a los aspectos identificados
- O4. Validar la usabilidad del simulador desarrollado con una muestra de estudiantes

1.2.3 Resultados esperados

Los resultados esperados del presente proyecto son:

- R1. Documento de análisis de requisitos con los aspectos del proceso de excavación arqueológica que deben ser incluidos en el simulador de excavaciones arqueológicas (O1)
- R2. Validación académica de los aspectos a ser incluidos en el simulador de excavaciones arqueológicas a través de evaluación por juicio experto de un profesional en la docencia de arqueología (O1)

R3.Documento de arquitectura de software que describa una que soporte la simulación de excavaciones arqueológicas (O2)

R4.Resultados de pruebas de estrés del simulador desarrollado con la arquitectura diseñada y que demuestre un mejor rendimiento que el simulador “El Pez Dorado” (O2)

R5.Módulo de simulación del proceso de excavación arqueológica (O3)

R6.Módulo de interfaz de usuario del simulador de excavación arqueológica (O3)

R7.Resultados de pruebas de usabilidad de la interfaz de usuario del simulador desarrollado (O4)

1.2.4 Mapeo de objetivos, resultados y verificación

A continuación, se resume la relación entre los objetivos específicos, los resultados específicos y los medios de verificación de estos en la tabla 1:

Tabla 1. Relación entre objetivos específicos, resultados esperados y medios de verificación (Elaboración propia).

Objetivo 1: Identificar aspectos sobre la excavación arqueológica que se puedan aplicar en un simulador		
Resultado	Meta física	Medio de verificación
R1. Documento de análisis de requisitos con los aspectos del proceso de excavación arqueológica que deben ser incluidos en el simulador de excavaciones arqueológicas	Documento	<ul style="list-style-type: none">• Cobertura de los requerimientos del simulador identificados en la entrevista con experto en docencia y de los aspectos del proceso de excavación arqueológica identificados en la problemática
R2. Validación académica de los aspectos a ser incluidos en el simulador de excavaciones arqueológicas a través de evaluación por juicio experto de un profesional en la docencia de arqueología con un	Encuesta	<ul style="list-style-type: none">• Aprobación de los aspectos considerados en el simulador por parte de profesional en la docencia de arqueología

mínimo de experiencia de 3 años		
Objetivo 2: Diseñar la arquitectura de software que soporte la simulación de excavaciones arqueológicas usando de modo eficiente los recursos del sistema		
R3. Documento de arquitectura de software que describa una que soporte la simulación de excavaciones arqueológicas	Documento	<ul style="list-style-type: none"> Existencia de documento de arquitectura de software Resultado aprobatorio de la validación de la arquitectura diseñada según la metodología a seguir Cobertura del 100% de los requerimientos del simulador identificados en R1
R4. Resultados de pruebas de estrés del simulador desarrollado con la arquitectura diseñada y que demuestre un mejor rendimiento que el simulador “El Pez Dorado”	Documento	<ul style="list-style-type: none"> Resultado aprobatorio tanto de las pruebas de estrés como de las pruebas comparativas del consumo de recursos computacionales respecto al simulador “El Pez Dorado”
Objetivo 3: Desarrollar un simulador de excavaciones arqueológicas en base a los aspectos identificados		
R5. Módulo de simulación del proceso de excavación arqueológica	Software, Documento	<ul style="list-style-type: none"> Cumplimiento del 100% de los requisitos identificados en R1 para la simulación del proceso de excavación arqueológica
R6. Módulo de interfaz de usuario del simulador de excavación arqueológica	Software, Documento	<ul style="list-style-type: none"> Cumplimiento del 100% de los requisitos identificados en R1 para la interfaz de usuario del simulador
Objetivo 4: Validar la usabilidad del simulador desarrollado con una muestra de estudiantes		

R7. Resultados de pruebas de usabilidad de la interfaz de usuario del simulador desarrollado	Documento	<ul style="list-style-type: none"> • Resultado aprobatorio de las pruebas con usuarios respecto a la usabilidad de la interfaz, según la metodología a utilizarse
---	-----------	--

1.3 Herramientas y métodos

En esta sección, se detallarán las herramientas y métodos más resaltantes que se utilizarán en el presente proyecto para alcanzar los resultados esperados descritos. En primer lugar, se presentará las relaciones entre los resultados esperados y las herramientas y métodos seleccionados para alcanzarlas. Después, se detallarán algunas de estas herramientas y métodos, según su relevancia para el presente proyecto.

1.3.1 Mapeo de resultados esperados y herramientas y métodos seleccionados

A continuación, se mostrarán las herramientas y métodos seleccionados para obtener cada uno de los resultados esperados descritos anteriormente. Esta información se muestra en la tabla 2.

Tabla 2. Mapeo de resultados esperados y herramientas y métodos seleccionados
(Elaboración propia)

Resultado	Medio de verificación	Herramientas y metodología
R1. Documento de análisis de requisitos con los aspectos del proceso de excavación arqueológica que deben ser incluidos en el simulador de excavaciones arqueológicas	<ul style="list-style-type: none"> • Cobertura de los requerimientos del simulador identificados en la entrevista con experto en docencia y de los aspectos del proceso de excavación arqueológica identificados en la problemática 	<ul style="list-style-type: none"> • Historias de usuario • Entrevista semi estructurada • Personas y Escenarios según integración de User-Centered Design y Extreme Programming
R2. Validación académica de los	<ul style="list-style-type: none"> • Aprobación de los 	<ul style="list-style-type: none"> • Entrevista

<p>aspectos a ser incluidos en el simulador de excavaciones arqueológicas a través de evaluación por juicio experto de un profesional en la docencia de arqueología con un mínimo de experiencia de 3 años</p>	<p>aspectos considerados en el simulador por parte de profesional en la docencia de arqueología</p>	<p>semiestructurada con experto en docencia de arqueología.</p> <ul style="list-style-type: none"> • Evaluación por juicio experto de los aspectos a ser incluidos en el simulador mediante un prototipado en papel (Rubin & Chisnel, 2008, pp. 18–19).
<p>R3. Documento de arquitectura de software que describa una que soporte la simulación de excavaciones arqueológicas</p>	<ul style="list-style-type: none"> • Existencia de documento de arquitectura de software • Resultado aprobatorio de la validación de la arquitectura diseñada según la metodología a seguir • Cobertura del 100% de los requerimientos del simulador identificados en R1 	<ul style="list-style-type: none"> • Modelo de vista de arquitectura de software “4+1” (Kruntschen, 1995) • Software Architecture Analysis Method
<p>R4. Resultados de pruebas de estrés del simulador desarrollado con la arquitectura diseñada y que demuestre un mejor rendimiento que el simulador “El Pez Dorado”</p>	<ul style="list-style-type: none"> • Resultado aprobatorio de las pruebas de estrés y comparación con desempeño del simulador “El Pez Dorado” 	<ul style="list-style-type: none"> • Lenguaje de programación C++ • Prueba de estrés y comparación de consumo promedio recursos computacionales
<p>R5. Módulo de simulación del proceso de excavación arqueológica</p>	<ul style="list-style-type: none"> • Cumplimiento del 100% de los requisitos identificados en R1 	<ul style="list-style-type: none"> • Lenguaje de programación C++ • Integración de User-

	para la simulación del proceso de excavación arqueológica	Centered Design con Extreme Programming
R6. Módulo de interfaz de usuario del simulador de excavación arqueológica	<ul style="list-style-type: none"> Cumplimiento del 100% de los requisitos identificados en R1 para la interfaz de usuario del simulador 	<ul style="list-style-type: none"> Lenguaje de programación C++ Integración de User-Centered Design con Extreme Programming
R7. Resultados de pruebas de usabilidad de la interfaz de usuario del simulador desarrollado	<ul style="list-style-type: none"> Resultado aprobatorio de las pruebas con usuarios respecto a la usabilidad de la interfaz, según la metodología a utilizarse 	<ul style="list-style-type: none"> Integración de User-Centered Design con Extreme Programming

1.3.2 Integración de metodología *Extreme Programming* y *User-centered design*

Para la obtención de requisitos, desarrollo y evaluación del presente proyecto, se ha optado por hacer uso de la integración de las herramientas brindadas por *User-centered design* y la metodología ágil *Extreme Programming*, según el trabajo de Aguilar y Zapata (Aguilar & Zapata, 2017). En primer lugar, debido a que se tiene como objetivo la usabilidad del software a desarrollarse, se ha considerado conveniente utilizar las herramientas descritas en la disciplina *User-centered design*. Esta disciplina coloca al usuario final como el centro del diseño, de manera que el producto desarrollado pueda ser utilizado correctamente por este y con el mínimo esfuerzo para aprender a utilizarlo (McLoone, Jacobson, Hegg, & Johnson, 2010, p. 2). Por otra parte, debido a que se harán pruebas de usabilidad con usuarios que podrían modificar el diseño inicial, se ha considerado también conveniente hacer uso de una metodología ágil. Esto se debe a que esta clase de metodologías no requiere que los requisitos del diseño se establezcan desde un inicio, sino que se adapta a los cambios de estos (Akpata & Riha, 2004, p. 1). Respecto a la metodología ágil escogida, se seleccionó a *Extreme Programming* debido a que esta metodología puede ser trabajada por un único desarrollador, como es el caso

de este proyecto, y aún mantener entre el 80% y 90% de sus beneficios (Akpata & Riha, 2004, p. 6).

A continuación, se presentarán breves descripciones de *User-centered design* y *Extreme Programming* para luego describir la integración de estos en la metodología a utilizarse.

1.3.2.1 User-centered design

User-centered design (UCD) es una disciplina que agrupa un conjunto de técnicas, procesos, métodos y procedimientos para diseñar productos usables y que coloca al usuario al centro de este diseño (Rubin & Chisnel, 2014). Los principios básicos que rigen esta disciplina son los siguientes:

a. Enfoque temprano en usuarios y sus tareas

El enfoque del diseño hacia el usuario debe darse durante todo el ciclo de vida del producto a desarrollarse, de manera sistemática y estructurada (Rubin & Chisnel, 2014).

b. Evaluación y medición del uso del producto

Se debe hacer énfasis en la medición de la facilidad de aprendizaje y facilidad de uso del producto, desde el diseño inicial hasta el desarrollo de este (Rubin & Chisnel, 2014).

c. Diseño iterado

El diseño debe ser desarrollarse a través de iteraciones a lo largo de cada etapa del ciclo de vida del producto, permitiendo repensar y rediseñarlo en cualquier etapa de esta (Rubin & Chisnel, 2014).

Las ventajas de hacer uso de esta disciplina incluye el desarrollo de productos más eficientes, efectivos y seguros; obtención de productos que requieren menor rediseño y que se integran más rápidamente; y la generación de soluciones más creativas respecto a los problemas de diseño, entre otras ventajas (McLoone et al., 2010, p. 11). A continuación, se listan algunas de las técnicas que incluye esta disciplina a lo largo del ciclo de vida de los productos desarrollados (Rubin & Chisnel, 2014).

- Investigación etnográfica
- Diseño participativo
- Investigación a través de *focus groups*
- Encuestas

- Pensando en voz alta
- Clasificación de tarjetas abierta y cerrada
- Prototipado en papel
- Evaluaciones heurísticas o expertas
- Pruebas de usabilidad
- Estudio de seguimiento
- Entrevistas

1.3.2.2 Extreme Programming

Extreme Programming, también conocido como XP, es una metodología ligera para equipos de cualquier tamaño que desarrollan software con requerimientos vagamente definidos o cambiantes (Beck & Andres, 2004). Esta metodología tiene como objetivos el desarrollo de software excepcional a un menor costo, menor cantidad de defectos y mayor productividad (Beck & Andres, 2004). Se le considera “extrema” en el sentido de que toma múltiples “mejores prácticas” de desarrollo de software y las lleva a sus extremos lógicos (Akpata & Riha, 2004). Ejemplo de este extremo lógico es no solo requerir la buena práctica de tener una comunicación con el usuario a lo largo del desarrollo, sino que recomienda que este debe trabajar en la misma habitación que los programadores e interactuar con ellos según sea necesario (Akpata & Riha, 2004).

En la tabla 3 se describen brevemente las doce prácticas principales sobre las que se basa Extreme Programming.

Tabla 3. Doce prácticas principales de Extreme Programming. Adaptado de (Akpata & Riha, 2004).

Práctica	Descripción
El juego de planeación	El equipo de negocio y de desarrollo escogen las funcionalidades del sistema que brindarán mayor valor al negocio.
Pequeños lanzamientos	Las funcionalidades deben ser puestas en producción temprano y actualizadas en periodos de tiempo corto.
Metáfora	El proyecto debe tener nombres y descripciones que permita el desarrollo y comunicación entre todos los involucrados.
Diseño simple	Siempre debe utilizarse el diseño más sencillo que cumpla los requerimientos actuales, sin preocuparse por los

	futuros cambios de requerimientos.
Pruebas	El software debe ser validado en todo momento, a través del desarrollo de pruebas antes del desarrollo de las mismas funcionalidades.
Refactorización	Mejora del código existente a través de la introducción de pequeños cambios a la estructura de este que lo mejoren sin cambiar su comportamiento
Programación en pares	El desarrollo del código debe darse a través de dos programadores utilizando un solo computador. Esto permite que el código sea desarrollado y, al mismo tiempo, validado.
Propiedad colectiva del código	El código debe pertenecer a todo el equipo de desarrollo y no pueden haber partes de este que le pertenezcan a un solo individuo. De esta manera, se fomenta la contribución de nuevas ideas por parte de cualquier miembro del equipo.
Integración continua	El software debe ser construido e integrado, como mínimo, una vez al día, tras haber pasado todas las pruebas.
Semana de 40 horas	Se debe tener una semana laboral de 40 horas para mantener la productividad y evitar la fatiga.
Usuario en el sitio	Uno o más usuarios deben localizarse y trabajar con el equipo de desarrollo, de manera que se tenga una comunicación efectiva entre ambas partes.
Estándares de programación	El equipo de desarrollo debe utilizar estándares de programación, de manera que se facilite la programación en pares y la propiedad colectiva del código.

1.3.2.3 Metodología a utilizarse

La metodología a utilizarse es una integración de las técnicas de UCD con el proceso de desarrollo de XP, basado en el trabajo de Aguilar y Zapata (Aguilar & Zapata, 2017). Esta integración permite obtener un proceso ágil y basado en usabilidad, al complementar el nivel de usabilidad proporcionado por XP con técnicas de UCD (Aguilar & Zapata, 2017). Es necesario resaltar que el proceso de XP contemplado solo incluye algunas de las prácticas de XP descritas, dado que el presente proyecto se desarrolla de manera individual. De esta manera, la integración de técnicas de usabilidad con el proceso de XP que se realizará de la siguiente manera:

- Evaluaciones con usuarios como pruebas unitarias al final de cada iteración de desarrollo. De manera específica, se utilizarán pruebas sumativas como método de evaluación (Rubin & Chisnel, 2014, p. 34).
- Método de “Personas” y “Escenarios” para definir historias de usuario
- Entrevistas semi estructuradas para obtener información sobre el usuario final
- Evaluación con usuarios para evaluar el producto final desarrollado. De manera específica, se utilizará una prueba de verificación (Rubin & Chisnel, 2014, p. 35)

La figura 1 muestra el proceso a seguirse en este proyecto. Los recuadros azules pertenecen al proceso de desarrollo de XP original y los recuadros rojos representan las técnicas de UCD añadidas al proceso.

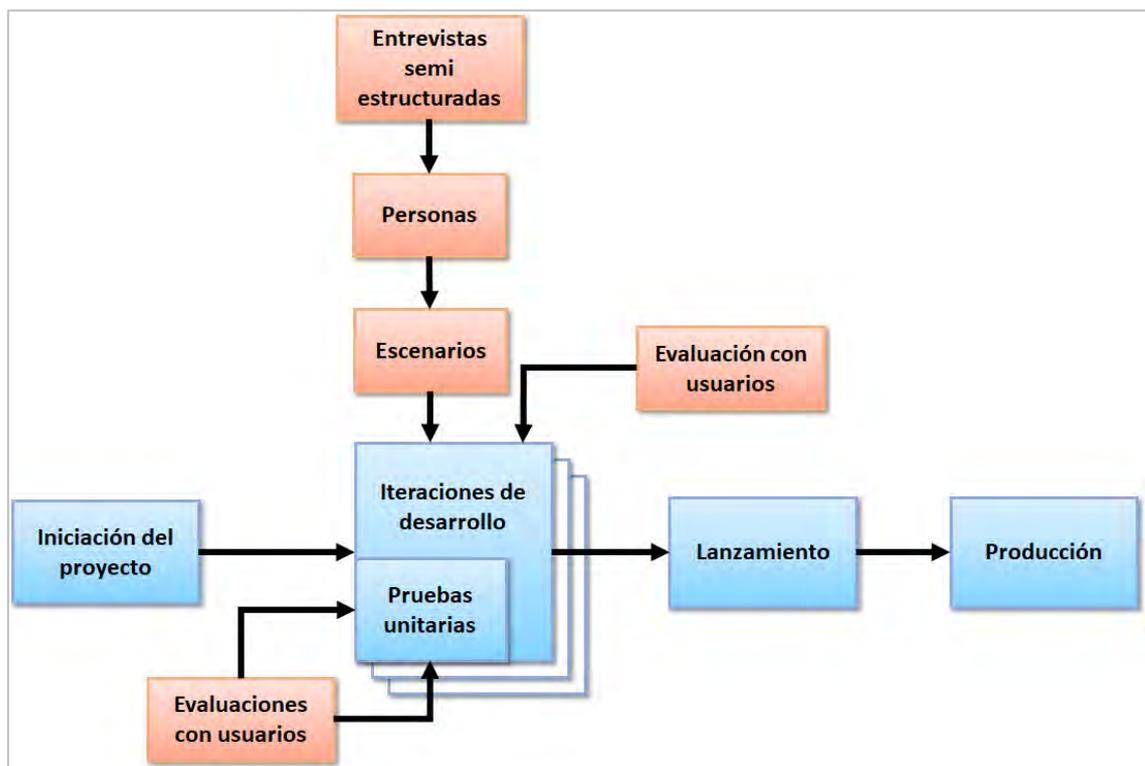


Figura 1. Proceso XP integrado con técnicas UCD. Adaptado de (Aguilar & Zapata, 2017)

1.3.3 Software Architecture Analysis Method

Como parte del objetivo específico 2, se diseñará una arquitectura de software que soporte la simulación de excavaciones arqueológicas que use de modo eficiente los recursos del sistema. Por este motivo, se ha decidido hacer uso del método “Software Architecture Analysis Method” (SAAM) para evaluar si se ha logrado diseñar una arquitectura de software que cumpla con el requerimiento de rendimiento.

El método SAAM utiliza un enfoque cualitativo basado en escenarios o contextos para determinar si el sistema evaluado logra cumplir los atributos de calidad deseados (Kazman, Abowd, Bass, & Clements, 1996, p. 1). La evaluación se realiza a través de cinco actividades, según se explican brevemente a continuación.

1. Descripción de la arquitectura candidata

Para el análisis, se requiere que la arquitectura de software esté descrita de una manera que todas las partes involucradas en el análisis puedan entender sus componentes de procesamiento y sus componentes de almacenamiento de datos (Kazman et al., 1996, p. 3).

2. Desarrollo de escenarios

Se deben desarrollar escenarios muestren las actividades que el sistema debe soportar actualmente y ante futuros cambios (Kazman et al., 1996, p. 3). Asimismo, los escenarios deben ser considerados desde la perspectiva de múltiples roles, no solo desde la perspectiva del usuario final (Kazman et al., 1996, p. 3).

3. Evaluación de cada escenario

Para cada escenario obtenido en la actividad anterior, se deberá analizar si el sistema actualmente permite soportarlo o requiere cambios en su arquitectura para hacerlo (Kazman et al., 1996, p. 3). Según este criterio, se clasificarán los escenarios en directos, si el sistema actual permite soportarlo, o en indirectos, en caso el sistema requiera de modificaciones para soportarlo (Kazman et al., 1996, p. 3). Adicionalmente, para cada escenario indirecto, se deberá indicar un listado de las modificaciones necesarias en la arquitectura de software para poder soportarlo (Kazman et al., 1996, p. 3).

4. Obtener interacciones entre escenarios

Se considera que dos escenarios interaccionan si ambos son clasificados como indirectos y requieren cambios en los mismos componentes de la arquitectura (Kazman et al., 1996, p. 3). De esta manera, la cantidad de interacciones entre escenarios mide el grado de cohesión de los componentes de una arquitectura y, por ello, posiblemente el número de futuros defectos del producto final (Kazman et al., 1996, p. 4). Por esta razón, el diseñador debe detectar las interacciones y corregir este problema (Kazman et al., 1996, p. 4).

5. Ponderar escenarios y las relaciones entre estos

Una vez evaluado cada escenario respecto a la arquitectura y su grado de

interacción, se deberán ponderar estos resultados, según la importancia que le de la organización a cada escenario, para obtener un puntaje final de la arquitectura evaluada (Kazman et al., 1996, p. 4).

Se seleccionó este método de evaluación de arquitecturas de software debido a que permite evaluaciones respecto a varias opciones de atributos de calidad, como mantenibilidad, modularidad, rendimiento, etc. (Mattsson, Grahn, Mårtensson, Box, & Ronneby, 2006, p. 8). Adicionalmente, también se seleccionó SAAM debido a que permite evaluar individualmente un diseño de arquitectura y no requiere, como otros métodos, de una comparación con múltiples diseños de arquitectura al mismo tiempo (Zbick, 2017, p. 5).

1.3.4 Modelo de vista de arquitectura de software “4+1”

“4+1” es un modelo que permite describir la arquitectura de sistemas basados en software (Kruntschen, 1995). Esta descripción se obtiene al hacer uso de múltiples vistas del sistema, según el punto de vista de múltiples perfiles que interactúan con el sistema, como usuarios finales, desarrolladores, administradores del proyecto, etc. (Kruntschen, 1995). Las vistas propuestas por este modelo y los diagramas que se utilizarán en cada una se muestran en la tabla 4. Para la elaboración de los diagramas indicados en la tabla 4, se hará uso de la notación indicada por Unified Modeling Language (Rumbaugh, Jacobson & Booch, 2005).

Tabla 4. Mapeo de vistas y diagramas utilizados (Elaboración propia).

Vista	Diagrama
Vista lógica	Diagrama de clases
Vista de desarrollo	Diagrama de componentes
Vista de procesos	Diagrama de actividades
Vista física	Diagrama de despliegue
Escenarios	Casos de uso

Para el presente proyecto, solo se hará uso de las siguientes vistas: vista lógica, vista

de desarrollo, vista de procesos y vista física. La vista “Escenarios” o “Casos de uso” ha sido excluida del presente proyecto debido a que esta vista representa una abstracción de los requerimientos más importantes (Kruntschen, 1995, p. 9), los cuales habrán sido definidos a través de historias de usuario de acuerdo al resultado esperado 1. Por ello, al tener definidas historias de usuarios, no se considera necesario volver a representar los requisitos más importantes mediante la elaboración de casos de usos.

1.3.5 Lenguaje de programación C++

Para el desarrollo de los módulos que componen el simulador, se eligió el lenguaje de programación C++ como herramienta de desarrollo. C++ es un lenguaje de programación de propósito general que es adecuado para el desarrollo de aplicaciones con recursos computacionales limitados (Stroustrup, 2013, p. 6). Se eligió este lenguaje de programación por las siguientes razones:

➤ Menor consumo de recursos computacionales

Dado que el problema de estabilidad identificado en el simulador “El Pez Dorado” se debió a un gran consumo de recursos computacionales, se ha decidido conveniente utilizar un lenguaje de programación que se caracterice por un bajo consumo de estos. Así, se escogió C++ dado que ha tenido el mejor ahorro de memoria RAM frente a otros lenguajes de programación en múltiples comparaciones (Abdo & Alali, 2016, p. 328; Prechelt, 2000, p. 27). Así también, este lenguaje de programación ha tenido un consumo de CPU dentro del promedio frente a otros lenguajes de programación (Abdo & Alali, 2016, p. 329). De esta manera, se espera que el software a desarrollarse con este lenguaje de programación pueda tener un menor consumo de recursos computacionales a que si se desarrollase con otro lenguaje de programación, como Java o C#.

➤ Mayor control sobre la arquitectura

Otra opción para desarrollar el simulador, según se encontró en el estado del arte, es hacer uso de programas existentes, como Matlab, para simular el comportamiento deseado (Jánó, Fodor, & Ilieş, 2016; Ruusuvoori et al., 2007; Soares & Casanova Alcalde, 2006). Sin embargo, dado que se ha reconocido, en la problemática, que la arquitectura de software utilizada puede afectar negativamente al rendimiento del simulador, se ha decidido no hacer uso de programas existentes. De esta manera, se tendrá mayor control sobre la

arquitectura de software utilizada y, por ende, también se tendrá mayor control sobre el rendimiento del simulador.

1.3.6 Prueba de estrés y comparación de consumo promedio de recursos computacionales

Tal como se describió en la problemática, la causa de la poca estabilidad del simulador “El Pez Dorado” se debió a un gran consumo de recursos computacionales. Por ello, el presente proyecto tiene como objetivo desarrollar un simulador de excavación arqueológica que tenga un menor consumo de recursos computacionales que el simulador anteriormente descrito. De esta manera, para determinar si se ha cumplido este objetivo, se utilizará la metodología de comparación utilizada en un trabajo de comparación de simuladores similar (Ur Rehman Khan, Bilal, & Othman, 2013). A partir de este, se compararán el consumo de memoria y uso de tiempo del procesador. Asimismo, se evaluará la característica de escalabilidad del simulador desarrollado.

Con el objetivo de comparar el consumo de los recursos mencionados, se realizarán mediciones al hacer uso de los simuladores por separado. Posteriormente, se realizará una experimentación numérica para comprobar el mayor o menor consumo de los recursos. Respecto al computador donde se realizarán las pruebas, este tiene las siguientes especificaciones:

- Sistema operativo: Windows 10 64 bits
- Procesador: Intel Core i7-6500U 2.50GHz
- Memoria RAM: 8.00 GB

1.4 Alcance, limitaciones y riesgos

1.4.1 Alcance

El presente proyecto de fin de carrera es un desarrollo tecnológico que tiene como finalidad facilitar la enseñanza del conocimiento práctico de la arqueología a través del desarrollo de un simulador educativo de excavación arqueológica para educación superior. Dado que el simulador tiene una finalidad académica, se busca que el simulador pueda ser utilizado por los alumnos de educación superior y que su uso no se encuentre restringido por la plataforma para la cual se encuentra desarrollado, como computadoras de escritorio o dispositivos móviles. De esta manera, se ha considerado conveniente orientar el desarrollo del simulador a la plataforma más popular en el desarrollo de simuladores, según la revisión del estado del arte: computadoras de escritorio. Así, se espera abarcar la mayor cantidad de alumnos de educación superior

sin que la plataforma seleccionada represente una limitación para su uso. Por otra parte, respecto al desarrollo del simulador, este se dividirá en dos módulos: módulo de simulación del proceso de excavación arqueológica y módulo de interfaz de usuario del simulador.

El primer módulo del desarrollo permitirá simular las actividades relacionadas a la excavación arqueológica, a ser definidas según el objetivo específico 1, y la gestión de los siguientes recursos del proceso de excavación: tiempo, personal, dinero y herramientas de excavación. Tanto las actividades relacionadas a la excavación como la gestión de recursos del proyecto estarán definidas por el simulador, por lo que el usuario final no podrá agregar, modificar u eliminar ninguna actividad. Además, respecto al proceso a simularse, solo se incluirán las etapas de prospección arqueológica, excavación arqueológica y el registro de artefactos. Otras etapas de los proyectos de excavación arqueológica, como la preparación del proyecto o el análisis de los artefactos encontrados, no serán consideradas en la simulación a desarrollarse. Asimismo, debido a que las actividades del proceso de excavación arqueológica difieren según el ambiente donde se realice, se ha escogido un ambiente desértico, basado en el que tiene “El Pez Dorado”, y este se mantendrá constante a lo largo de la simulación.

El segundo módulo del desarrollo brindará una interfaz de usuario que le permita al usuario hacer uso del simulador desarrollado. Este módulo no hará uso de instrumentos de realidad aumentada o virtual, sino mediante gráficos 2D. Dado que el presente proyecto se trata de un proyecto de fin de carrera de la carrera de ingeniería informática, no se considerará la creación estos gráficos como parte del presente proyecto. En su lugar, se utilizarán los gráficos utilizados por el simulador “El Pez Dorado”.

Para el desarrollo y validación final de simulador, se realizarán pruebas de usabilidad mediante pruebas con usuario, según la metodología descrita en el acápite 1.3.1 del presente proyecto. Cada una de estas pruebas se realizarán con tres (3) estudiantes de pregrado de la PUCP, según la cantidad utilizada originalmente por los autores de la metodología utilizada (Aguilar & Zapata, 2017, p. 7) y debido a que incrementar la cantidad de estudiantes necesarios incrementaría el riesgo de no realizar las pruebas con usuarios por no contar con la cantidad mínima de alumnos requeridos.

1.4.2 Limitaciones

Para el presente proyecto, se han identificado las siguientes limitaciones:

- **Participación de profesional en docencia de arqueología.** Es necesaria su colaboración para realizar la validación académica de los requisitos identificados como parte del resultado esperado 1.
- **Participación de tres (3) estudiantes de pregrado de la PUCP.** Es necesaria su colaboración para realizar pruebas con usuarios tanto en cada iteración de desarrollo como en la evaluación de usabilidad del producto final.

1.4.3 Riesgos

En la tabla 5 se muestran los riesgos identificados, la probabilidad de ocurrencia, el impacto estimado y las medidas que se adoptarán para mitigar el riesgo.

Tabla 5. Mapeo de riesgos, impacto estimado y medidas adoptadas (Elaboración propia).

Riesgo	Probabilidad	Impacto estimado	Medidas adoptadas
No contar con la participación de un profesional en docencia de arqueología	Medio	Alto	<ul style="list-style-type: none"> ➤ Obtener el compromiso del profesional antes de iniciar la ejecución del proyecto ➤ Identificar otros posibles candidatos en caso pierda el compromiso del profesional inicial
No contar con la cantidad requerida de alumnos de pregrado de la PUCP	Bajo	Alto	<ul style="list-style-type: none"> ➤ Ofrecer recompensas por la participación en la validación de usabilidad en caso no se consiga la participación del mínimo de alumnos
Incorrecta estimación de los tiempos requeridos en cada actividad del proyecto	Medio	Medio	<ul style="list-style-type: none"> ➤ Considerar que cada actividad tomará un 15% de tiempo extra como tiempo de respaldo
Pérdida del código fuente utilizado	Bajo	Alto	<ul style="list-style-type: none"> ➤ Utilizar un sistema de control de versiones para gestionar las versiones del simulador desarrollado
No se cuente con los	Bajo	Alto	<ul style="list-style-type: none"> ➤ Identificar en internet fuentes de

gráficos suficientes para realizar la simulación del proceso.			gráficos relacionados a la arqueología que puedan ser utilizados libremente en el presente proyecto.
---	--	--	--

1.1 Viabilidad

A continuación, se realizará un análisis de la viabilidad del presente proyecto desde tres perspectivas: viabilidad técnica, viabilidad temporal y viabilidad económica.

1.1.1 Viabilidad técnica

Se considera que no existen limitaciones técnicas para conseguir los resultados esperados propuestos para cada objetivo específico. Esto se debe a las razones expuestas en la tabla 6:

Tabla 6. Razones de viabilidad técnica por objetivo específico (Elaboración propia).

Objetivo específico	Razón de viabilidad
O1. Identificar aspectos sobre la excavación arqueológica que se puedan aplicar en un simulador	<ul style="list-style-type: none"> ➤ Se cuenta con experiencia en la identificación de requerimientos de un software, de acuerdo a lo aprendido a lo largo de la carrera de ingeniería informática. ➤ Se posee extensa bibliografía, tanto física como virtual, de dónde obtener información sobre el proceso de excavación arqueológica.
O2. Diseñar la arquitectura de software que soporte la simulación de excavaciones	<ul style="list-style-type: none"> ➤ Se cuenta con experiencia en el diseño de arquitectura para sistemas interactivos de alto rendimiento, según el trabajo desarrollado por el autor en el curso “Lenguajes de programación 2”. ➤ Se posee bibliografía sobre las

	mejores prácticas para diseñar una arquitectura de software de alto rendimiento.
O3. Desarrollar un simulador de excavaciones arqueológicas en base a los aspectos identificados	➤ Se cuenta con experiencia en el desarrollo de software interactivos similares, según el trabajo desarrollado por el autor en el curso “Lenguajes de programación 2”.
O4. Validar la usabilidad del simulador desarrollado con una muestra de estudiantes	➤ Se posee bibliografía que describe a detalle la forma correcta de realizar la validación de usabilidad, según la metodología descrita en el acápite 1.3.

1.1.2 Viabilidad temporal

En el anexo C se puede encontrar el cronograma de actividades del proyecto, tanto desde la planificación como la ejecución de este. En este documento se encuentran las actividades a desarrollarse en el proyecto, la duración estimada y las fechas de comienzo y fin de cada una. Las actividades descritas tienen una duración de 370 días si se toman de manera individual. Sin embargo, algunas de ellas consideran una ejecución simultánea, por lo que el tiempo total se reduce. De esta manera, se obtiene que el proyecto tendrá una duración de aproximadamente 6 meses, por lo que es viable temporalmente desarrollarlo entre los ciclos académicos 2018-1 y 2018-2.

Cabe resaltar que el tiempo de ejecución del proyecto es un estimado, debido a que los requerimientos del simulador aún serán definidos como parte del objetivo específico 1. Por ello, se ha estimado el tiempo a partir del alcance que ha tenido el simulador “El Pez Dorado”, el cual también se basa en los aspectos del proceso de excavación arqueológica.

1.1.3 Viabilidad económica

A partir del cronograma de actividades, descrito en el anexo C, se obtiene que se necesitarán 370 días para ejecutar las actividades del proyecto. Utilizando una estimación de 5 horas-hombre por día planificado, se obtiene que para ejecutar el

presente proyecto se necesitarían costear 1850 horas-hombre. Por otra parte, también se debe considerar las horas-hombre invertidas por la participación de terceros. Considerando la participación del profesional en la docencia y a los 3 alumnos que realicen las pruebas con usuario, se necesitarían, adicionalmente, otras 50 horas-hombre. De esta manera, suponiendo un costo de 20 soles por hora-desarrollador y 150 soles por hora-externo, promediando el costo del experto y de los usuarios, se obtiene un costo total de 44,500 soles por horas invertidas. Sin embargo, no se incurrirán en estos costos durante la ejecución del proyecto ya que el único desarrollador es el autor y a que la participación de los terceros identificados es voluntaria.

Por otro lado, se podría materializar el segundo riesgo identificado en la tabla 5. De darse este caso, se estima que la inversión requerida para las recompensas sería de aproximadamente 60 soles y sería asumido en su totalidad por el autor del presente proyecto. Es necesario resaltar que el autor tiene la capacidad económica para asumir el costo identificado, por lo que no afecta la viabilidad económica del proyecto.

De esta manera, con la descripción de los costos y las situaciones en las que se asumirían, se concluye que el proyecto es viable económicamente.



Capítulo 2. Marco conceptual

En este capítulo se definirán los conceptos utilizados en la problemática descrita del presente proyecto. Asimismo, también se explicará, según sea necesario, el contexto en el cual se aplican estos conceptos en el presente documento.

2.1 Arqueología

Se define como el estudio de la historia del hombre a través de los restos materiales (Society for American Archaeology, n.d.-b). Este campo de estudio forma parte de la antropología, definida como el estudio del hombre, y se especializa en el estudio e interpretación del pasado del hombre (Renfrew & Bahn, 2007, p. 9).

La arqueología puede clasificada de diversas maneras, como por los periodos históricos que analiza o el hábitat natural donde se desempeña (Renfrew & Bahn, 2007, p. 11). Para los fines del presente proyecto, se utilizará la clasificación de la arqueología entre teoría y práctica de esta (Johnson, 2010, p. 3). Asimismo, se dará un énfasis en este último tipo de arqueología, en especial en los conocimientos necesarios para poder ejercer la práctica arqueológica, como la excavación arqueológica.

2.2 Restos materiales

Objetos orgánicos, como huesos, plantas y suelos, o inorgánicos, como herramientas o edificaciones construidos, modificados o utilizados por el hombre en el pasado (Renfrew & Bahn, 2007, pp. 43–44). Para los arqueólogos, los restos materiales representan su objeto de estudio y es a través de estos que obtienen información sobre la vida del hombre en el pasado (Barker et al., 2016, p. 5). El estado de conservación de estos restos materiales depende tanto del trato que ha recibido por parte de las personas del pasado como de las condiciones naturales que los hayan podido afectar, como el suelo y el clima (Renfrew & Bahn, 2007, p. 63). Por ello, es importante la forma en la cual se interactúa con los restos materiales, la forma a través del cual se recuperan y la conservación a la cual se somete (Renfrew & Bahn, 2007, p. 63). En el presente proyecto, el énfasis que se hará será sobre el proceso de excavación arqueológica como forma de extracción y recuperación de los restos materiales.

2.3 Simulación

La simulación puede definirse como la imitación del comportamiento de un sistema o proceso del mundo real a lo largo del tiempo (Banks, Nelson, Carson, & Nicol, 2004, p. 3). Esta imitación del comportamiento se logra a través de un conjunto de suposiciones,

las cuales son expresadas de forma matemática, lógica o mediante relaciones entre entidades (Banks et al., 2004, p. 3). Una vez que se ha validado que el comportamiento simulado es el esperado, se puede utilizar la simulación para determinar el comportamiento del sistema ante cambios específicos y el impacto de estos (Banks et al., 2004, p. 3). Un caso especial de simulación es la emulación, en la cual la imitación realizada sobre un sistema debe ser realista (Banks et al., 2004, p. 8). Es decir, en una emulación no se realizan cambios sobre el sistema, sino que se simula el comportamiento real de este.

Un ejemplo de uso de simulación es el simulador de vuelo *Avion Flight Simulator 2015* (Seven Summits Studio, 2015). Este producto simula el comportamiento del proceso de pilotear un avión mediante condiciones ideales, incluyendo la interacción con pistas de aterrizajes, cabinas de pilotaje e, incluso, los efectos de diversos tipos de climas sobre el proceso (Seven Summits Studio, 2015). Este simulador también puede ser utilizado como un emulador, ya que se puede configurar para que brinde una experiencia de vuelo real sin hacer uso de condiciones de vuelo ideales (Seven Summits Studio, 2015). En la figura 2 se muestra el simulador desde la vista de la cabina del piloto.



Figura 2. Vista de la cabina de piloto del simulador. Imagen tomada de (Seven Summits Studio, 2015)

En el presente proyecto, tal como en el ejemplo mostrado, se desarrollará una simulación del proceso de excavación arqueológica mediante el uso de software. Sin

embargo, cabe resaltar que el simulador a desarrollarse no realizará una emulación del proceso de excavación, dado que no imitará el proceso de manera realista sino solo con las actividades que se identificarán y seleccionarán como parte del objetivo específico 1.

2.4 Usabilidad

El término *usabilidad* puede ser definido como el grado en el cual un producto puede ser utilizado por un usuario en específico para alcanzar objetivos específicos con efectividad, eficiencia y satisfacción en un contexto específico de uso (International Organization for Standardization, 1998). De esta manera, se entiende que el producto permite alcanzar los objetivos con rapidez, se comporta de la manera esperada por el usuario y que es percibido de una manera positiva por el usuario al ser utilizado (Rubin & Chisnel, 2014, p. 4).

En el presente proyecto, se plantea el desarrollo de un simulador y que este sea validado como usable, según la definición brindada, por los usuarios que lo utilizarán. La metodología a seguirse para validar la usabilidad del simulador se explica en el acápite 1.3.1.

2.5 Excavación arqueológica

Proceso principal para la obtención del objeto de estudio de la arqueología, es decir, los restos materiales (Renfrew & Bahn, 2007, p. 513). Debido a sus altos costos y al peligro de destrucción de restos materiales, este proceso debe ser justificado mediante un análisis previo del terreno a excavar (Renfrew & Bahn, 2007, p. 96). Una vez justificada la elección del terreno a excavar, este proceso continúa con la remoción de tierra, la recuperación de los restos materiales encontrados y el adecuado registro y clasificación de estos (Renfrew & Bahn, 2007, p. 105). Una vez recuperados los restos materiales a través de este proceso, estos podrán ser analizados e interpretados posteriormente para entender a las sociedades humanas que ocuparon el sitio escogido.

Capítulo 3. Estado del arte

El objetivo de este capítulo es identificar y dar a conocer el estado del arte de los simuladores educativos utilizados o desarrollados en diferentes rubros, no solo simuladores relacionados a la arqueología. Asimismo, una vez identificados los simuladores utilizados o desarrollados en otros rubros, se tiene como sub objetivo conocer las tecnologías que se utilizan o han utilizado para desarrollar los simuladores identificados. De esta manera, se espera comprender las razones por las que se utilizaron aquellas tecnologías y evaluar, posteriormente, si estas deben ser utilizadas en el presente proyecto. Por último, también se tiene como sub objetivo conocer acerca de cuáles son las plataformas para las cuales se han desarrollado los simuladores educativos. De esta manera, se delimitará posteriormente el enfoque del desarrollo del simulador del presente proyecto a la plataforma más popular, donde se espera tener mayor acogida por los estudiantes.

Como metodología para realizar la revisión del estado del arte, se ha escogido la revisión sistemática. Esta decisión se ha tomado debido a que esta revisión permite “identificar, evaluar e interpretar las investigaciones relevantes disponibles” (Kitchenham, 2004) de manera objetiva y reproducible sobre el tema de interés.

3.1 Protocolo de revisión

Pregunta de investigación

En el estado del arte, la formulación de las preguntas de investigación desempeña una de las fases más importantes, debido a que es a partir de esta que se basan el resto de fases de la investigación del estado del arte. Así, dada su importancia, se escogió utilizar el método PICOC para la correcta formulación de las preguntas. A continuación, se muestran los valores PICOC escogidos:

- *Population* o población - Proyectos de desarrollo o aplicación de simuladores
- *Intervention* o intervención - Utilización de simuladores en educación
- *Comparison* o comparación - No aplica.
- *Outcome* o salida - Proyectos donde se apliquen los simuladores a un espacio académico
- *Context* o contexto - El contexto es amplio. Se consideran todas las áreas donde se puedan aplicar simuladores educativos.

De esta manera, con los valores PICO escogidos se formularon las preguntas de investigación necesarias. Esta información se muestra en la tabla 7.

Tabla 7. Formulación de preguntas de investigación a partir de método PICO (Elaboración propia).

Población	Intervención	Salida	Contexto	Preguntas de investigación
Proyectos de desarrollo o aplicación de simuladores	Educación	Proyectos donde se apliquen simuladores a un espacio académico	Amplio. Considera todas las áreas de aplicación	P1 ¿Qué proyectos de desarrollo de simuladores o aplicaciones de estos se han realizado orientados a educación?

Adicionalmente, a partir de la pregunta de investigación obtenida, se han definido las siguientes subpreguntas de investigación:

- P1.1.** ¿Cuáles son los rubros para los cuales se han desarrollado simuladores orientados a educación?
- P1.2.** ¿A qué plataformas está destinado el uso de los simuladores orientados a educación?
- P1.3.** ¿Qué tecnologías se han utilizado para desarrollar simuladores orientados a educación?

Cadenas de búsqueda para cada pregunta

Tras formular la pregunta de investigación, se generaron las subcadenas para cada elemento del método PICOC. Con este fin, se realizó una búsqueda previa en la base de datos seleccionada para determinar las palabras claves que brinden conjunto de resultados con un nivel de detalle adecuado. De esta manera, se determinaron las siguientes subcadenas:

- Cadena población - C1: (project* OR application*) AND (simulator*) AND (development OR tool*)
- Cadena intervención - C2: education OR educational
- Cadena comparación - C3: No aplica
- Cadena salida - C4: (education OR educational) AND (simulator*)
- Cadena contexto - C5: No se considera para no restringir la búsqueda.

Debido a que la cadena C3 no aplica, que la cadena C4 está contenida en las cadenas C1 y C2 y que la cadena C5 se encuentra vacía, la cadena final de búsqueda determinada para la pregunta de investigación P1 fue:

C1 AND C2

Fuentes de información

A partir de las recomendaciones de la guía temática del sistema de bibliotecas PUCP, se decidió realizar la revisión sistemática en la base de datos *IEEE Xplore Digital Library*.

Criterios de inclusión y exclusión

Los criterios de inclusión y exclusión utilizados se describen a continuación en la tabla 8.

Tabla 8. Criterios de inclusión y exclusión utilizados (Elaboración propia).

Criterios de inclusión	Criterios de exclusión
CI1. Artículos que traten sobre el desarrollo u aplicación de un simulador virtual que se encuentre orientados a la educación.	CE1. Artículos escritos en un idioma distinto del inglés o español. CE2. Artículos publicados con una antigüedad mayor a 15 años.

Se ha considerado incluir el criterio de exclusión 2 debido a que, como parte de la subpregunta de investigación P1.3., se van a considerar las tecnologías utilizadas para desarrollar simuladores orientadas a educación. De esta manera, se considera necesario excluir simuladores que pudieran haber sido desarrollados con tecnología anticuada, según la fecha de publicación del artículo donde se presentan.

3.2 Revisión y discusión

En esta sección, se mostrarán los resultados de la ejecución del protocolo de búsqueda. Además, se analizarán las principales características de los documentos que sí fueron considerados en la revisión sistemática.

En primer lugar, tras realizar la búsqueda en la base de datos seleccionada con la cadena de búsqueda, se obtuvieron 646 documentos que cumplieran con los requisitos. Aplicando los criterios de inclusión y exclusión, todos estos documentos fueron clasificados según se describe en la tabla 9.

Tabla 9. Cantidad de documentos por clasificación (Elaboración propia).

Clasificación	Cantidad
Documentos que coinciden con la cadena de búsqueda	646
Documentos seleccionados según el criterio de inclusión	98
Documentos excluidos según el criterio de exclusión	25
Documentos duplicados	5
Documentos seleccionados final	68

En el anexo A se encuentra una lista de todas las publicaciones que fueron seleccionadas para este análisis, tras aplicar los criterios de búsqueda sistemática.

A partir de los resultados encontrados, se dará respuesta a la pregunta de investigación:

P1. ¿Qué proyectos de desarrollo de simuladores o aplicaciones de estos se han realizado orientados a educación?

Dado que no todos los simuladores revisados no tienen un nombre asignado, no se ha detallado una lista de los simuladores con este atributo. En su lugar, se provee, en el anexo A, la lista de publicaciones que tratan sobre los simuladores que corresponden a la pregunta de investigación, así como información adicional sobre las sub preguntas de investigación.

P1.1. ¿Cuáles son los rubros para los cuales se han desarrollado simuladores

orientados a educación?

Se ha encontrado que el desarrollo y aplicación de simuladores es una práctica académica extendida entre distintos rubros. Así, se identificó el uso de un simulador de pacientes virtual en medicina para entrenar a nuevos médicos sobre los procedimientos a seguir (Oladosu, Adigun, & Emuoyibofarhe, 2013). Asimismo, también se identificó el uso de simuladores para reproducir y aprender sobre el comportamiento de procesadores de computadora (Imai & Takeichi, 2015). Otros campos de estudio donde se encontró el uso de simuladores con fines académicos son la electrónica (Santos, Chibante, Araújo, & Carvalho, 2009), la aeronáutica (Arnaldo, Perez, Crespo, & Alonso, 2011), la medicina (Rozanek, Kudrna, Hozman, & Kopecky, 2013), entre otros (ver anexo A). Tal como estiman los autores de estos tres campos de estudio antes mencionados, se estima que el uso de simuladores educativos tiene un impacto positivo en el aprendizaje de los alumnos (Rozanek et al., 2013), aunque dos de los autores también afirman que mayores estudios hacen falta para confirmar este hecho (Arnaldo et al., 2011; Santos et al., 2009).

P1.2. ¿A qué plataformas está destinado el uso de los simuladores orientados a educación?

Respecto a las plataformas objetivo, se ha encontrado que se han desarrollado simuladores educativos orientados principalmente a ser utilizados como aplicaciones de escritorio, al representar estos el 75% del total desarrollados (Lambertsen, Tang, Davies, & Morecroft, 2017; Rauen, Rolfe, Kantarci, Banavar, & Freitag, 2016). Cabe resaltar que esta cifra considera los simuladores encontrados que no especifican una plataforma objetivo en específico (Huitt, 2003; Ruusuvoori et al., 2007), dado que se trabajó con el supuesto de que se orientan a esta plataforma. Después, la siguiente plataforma más utilizada han sido los navegadores web, con el 19% de los simuladores en esta categoría (Salama & Shawish, 2013; Tachikawa & Nakamura, 2013; Truchly, Danielovič, Dukát, & Pôbiš, 2016). Finalmente, la plataforma a la cual se orientan la menor cantidad de simuladores son los dispositivos móviles, con solo 4% del total de simuladores encontrados (Capuz, Policarpio, Quinto, Reyes, & Uy, 2016; Ghazali, Hassan, & Ali, 2013; Vega-Medina, Perez-Gutierrez, Tibamoso, Uribe-Quevedo, & Jaimes, 2014).

P1.3. ¿Qué tecnologías se han utilizado para desarrollar simuladores orientados a educación?

Debido a que no todos los simuladores encontrados han detallado las tecnologías utilizadas en su desarrollo, los porcentajes presentados en esta sección corresponden solo a los simuladores que sí detallaron su implementación. De esta manera, se ha encontrado que el lenguaje de programación más utilizado es Java, al ser utilizado por el 30% de los simuladores (Bacca, Molina, Cavanzo, & Guevara, 2009; Ghazali et al., 2013; Shaikh & Shahzad, 2017). El 70% de los casos que han utilizado esta herramienta, explican que este hecho se debe a que los programas desarrollados en este lenguaje de programación pueden ser fácilmente exportados a otras plataformas, de manera que permite la fácil difusión del simulador (García, 2011; Lee, Jeong, Park, Shin, & Lee, 2007; Wang, Sun, Du, & Jia, 2006). Los siguientes lenguajes de programación más utilizados pertenecen al framework .Net, con el 18% de los casos (Oztekin, Temurtas, & Gulbag, 2010; Stamatescu et al., 2017). Después, Javascript fue el siguiente lenguaje más utilizado, con el 16% de los simuladores (Noll et al., 2014; Paramita & Smitha, 2017). Cabe resaltar que el 100% de los simuladores orientados a plataformas web se desarrollaron con este lenguaje de programación. Finalmente, otros lenguajes con los cuales se desarrollaron dos o más simuladores son C++ (12%) y Actionscript (9%).

Respecto a herramientas de desarrollo, se encontraron simuladores que hicieron uso de software de simulación pre-existentes como parte de su proceso de desarrollo (ver anexo A). El único software que fue utilizado en dos o más casos fue Matlab, el cual representó el 35% de los casos (Jánó et al., 2016; Ruusuvoori et al., 2007; Soares & Casanova Alcalde, 2006).

3.3 Conclusiones

Respecto a la pregunta de investigación, se encuentra que los proyectos de desarrollo o aplicación de simuladores educativos son una práctica académica común en múltiples y variados campos de conocimiento, como medicina, redes de computadora e ingeniería electrónica. Además, se han encontrado testimonios donde se estima que el uso de simuladores en la enseñanza tiene un impacto positivo en los alumnos.

Por otro lado, se ha encontrado que las plataformas más comunes a la cual se orientan los simuladores educativos son los computadores de escritorio. Otras plataformas en las cuales se encuentran simuladores, aunque en menor cantidad, son los navegadores web y los dispositivos móviles. Así también, se han identificado las tecnologías más comunes para el desarrollo de simuladores orientados a educación. De estas, el lenguaje de programación más utilizado es Java, debido, en la mayoría de casos, a la

portabilidad que provee entre plataformas. Los siguientes lenguajes de programación más utilizados pertenecen al framework .Net. Después, se encontró que Javascript es la siguiente opción más utilizada, el cual es utilizado en el 100% de los simuladores orientados a plataformas web. Finalmente, se ha identificado a Matlab como la herramienta de simulación más utilizada para el desarrollo de los simuladores encontrados.



Capítulo 4. Identificación de los aspectos sobre la excavación arqueológica que puedan aplicar en un simulador

En el presente capítulo, se alcanzará el primer objetivo específico del presente proyecto de tesis. Para ello, se identifican los aspectos, mediante la formulación de historias de usuario, que deberá cubrir el alcance del simulador a desarrollarse. Estos requerimientos, como se detallará posteriormente, se formularon tras la identificación de usuarios y escenarios y tras la consulta de fuentes escritas y orales. Asimismo, de acuerdo a la metodología seguida, se validaron académicamente los requisitos obtenidos mediante una segunda entrevista con el experto en la docencia de arqueología. De esta manera, se obtuvo un documento de análisis de requisitos validado por una experta en la docencia de arqueología.

4.1 Formulación de historias de usuario con los aspectos del proceso de excavación arqueológica

De acuerdo a la metodología seguida, las historias de usuario se formularon a partir de la información recopilada a través de una entrevista semi estructurada con un experto en la docencia de arqueología y a la identificación de personas y escenarios. De esta manera, los pasos seguidos para la formulación de las historias de usuario se listan a continuación.

- i. Preparación de la entrevista con experto
- ii. Desarrollo de la entrevista con experto
- iii. Identificación de personas y escenarios
- iv. Formulación del catálogo de requisitos e historias de usuario

4.1.1 Preparación de la entrevista

Como parte de la metodología seguida, se realizó una entrevista con un experto en la docencia de arqueología. Esta entrevista tuvo como objetivo la obtención de información sobre las actividades a detalle que componen los procesos de prospección y excavación arqueológica. De esta manera, se recopilarían requerimientos iniciales que podrían considerarse en el proyecto. Para ello, se realizó una investigación previa sobre la ejecución de los procesos que conforman un proyecto arqueológico antes de llevar a cabo la entrevista. Así, el autor decidió por obtener información previa a partir de dos libros que componían la bibliografía del curso ARQ107 *Arqueología*, curso sobre el cual

se basó el simulador anterior, “El Pez Dorado”. Estos dos libros fueron *Arqueología: Teoría, Métodos y Práctica* (Renfrew & Bahn, 2007) y *Manual de campo del arqueólogo* (Ines, Burke, & Smith, 2007). Tras una lectura general de los libros, se identificó y leyó detenidamente los capítulos que explican a detalle los procesos de prospección y excavación arqueológica. De esta manera, se le dio un mayor énfasis al capítulo “Where? Survey and Excavation of Sites and Features” del primer libro y a los capítulos “La búsqueda de yacimientos” y “Técnicas básicas de excavación” del segundo libro. Asimismo, para complementar la información leída de los libros anteriormente mencionados, se experimentó con el simulador “El Pez Dorado” para visualizar las funcionalidades que abarcaba.

Respecto a la elección del experto en la docencia en arqueología, se escogió a la Dra. Francesca Fernandini, coordinadora de la especialidad de arqueología y directora del proyecto arqueológico en el sitio arqueológico “Cerro de Oro”. Se escogió a la Dra. Fernandini como experta por su experiencia en la docencia, su asociación con un proyecto arqueológico que se desarrollaba en un ambiente similar al que recrea el simulador “El Pez Dorado” y por su disposición para participar en el presente proyecto. De esta manera, a partir de su experiencia podría brindar información acerca de los procesos de prospección y excavación arqueológica, de acuerdo al objetivo de la entrevista. Así, para tener un mejor entendimiento de los casos prácticos que pudiera presentar la entrevistada, también se realizó una investigación sobre los últimos trabajos publicados por la Dra. Fernandini, ya sea a través de medios escritos, como publicaciones científicas, o audiovisuales, como entrevistas.

A partir de la información leída, las dudas que surgieron durante la lectura y de la experiencia de utilizar el simulador “El Pez Dorado”, se formuló la estructura sobre la que se basó la entrevista semi estructurada. Esta estructura puede ser encontrada en el anexo D.

4.1.2 Desarrollo de la entrevista

Una vez formulada la estructura de la entrevista semi estructurada, el siguiente paso para formular el catálogo de requisitos fue realizar la entrevista propiamente dicha con la Dra. Francesca Fernandini. La entrevista se realizó el día 13 de agosto de 2018 en su oficina en el departamento de Humanidades de la PUCP. La transcripción de la entrevista se puede encontrar en el anexo E del presente texto.

En la entrevista se detallaron las acciones que componen tanto el proceso de prospección como el proceso de excavación arqueológica. Es necesario resaltar que las acciones fueron detalladas tanto en un contexto genérico como en el contexto del proyecto arqueológico del sitio “Cerro de Oro”. Asimismo, también se brindó información acerca de la perspectiva de gestión de proyectos, al hablarse de las cantidades por tipo de trabajador contratado y cuáles son los factores limitantes para la contratación.

4.1.3 Identificación de personas y escenarios

De acuerdo a la integración de *Extreme Programming* y *User-centered design* descrita, se definirán personas y escenarios para obtener una mejor perspectiva de usabilidad en los requerimientos identificados. De esta manera, se definen los grupos de personas que utilizarán el simulador de excavaciones arqueológicas a continuación:

- **Grupo 1:** Alumnos de educación superior sin ninguna información acerca de los procesos de prospección y excavación arqueológica.
- **Grupo 2:** Alumnos de educación superior con información acerca de los procesos de prospección y excavación arqueológica.

Debido a que la finalidad de una persona con información sobre los procesos mencionados al utilizar el simulador será diferente a la finalidad de una persona sin información, se utilizó el criterio de información sobre arqueología que tiene el usuario para definir los grupos. En el primer caso, la finalidad del usuario es poner en práctica la teoría recibida y, en el segundo, la finalidad del usuario es recibir información inicial sobre los procesos.

De esta manera, a partir de los grupos de personas identificados, se definen los siguientes escenarios a tomar en consideración en las historias de usuario:

Tabla 10. Descripción del escenario 1 (Elaboración propia)

Escenario 1
<p>Diego: Alumno de ingeniería informática sin conocimientos previos de arqueología</p> <p>Diego se encuentra realizando sus estudios en una reconocida universidad peruana que cuenta con restos arqueológicos dentro de su campus principal. Como parte de su recorrido para llegar a su aula de clase, Diego pasa junto a un yacimiento</p>

arqueológico dentro de la universidad que se encuentra cercado por planchas de metal. Según le informan, arqueólogos de la misma casa de estudio se encuentran excavando el yacimiento. Debido a que Diego pasa junto al yacimiento a diario, siente curiosidad por el proceso de excavación y decide hacer uso del simulador de excavaciones arqueológicas.

Diego accederá a la aplicación e iniciará una nueva simulación, donde se le explicará, brevemente, qué etapas del proyecto arqueológico cubre el simulador. A continuación, Diego visualizará una explicación más detallada sobre en qué consiste la etapa de prospección arqueológica. Luego, Diego seleccionará el método de prospección a utilizar, junto con una explicación detallada sobre en qué consiste cada una. Después, indicará la cantidad de participantes en cada yacimiento encontrado para luego pasar a la etapa de excavación. Nuevamente, se le explicará a detalle en qué consiste esta etapa. Finalmente, cuando Diego encuentre todos los artefactos en los yacimientos excavados, finalizará la simulación y volverá al menú principal. De esta manera, Diego pudo incrementar sus conocimientos acerca de los procesos de prospección y excavación arqueológica.

Tabla 11. Descripción del escenario 2 (Elaboración propia)

Escenario 2
<p>Gina: Alumna de la carrera de arqueología</p> <p>Gina acaba de terminar de recibir sus clases teóricas respecto al módulo de trabajo de campo como parte de su carrera. Sin embargo, a ella le gustaría poner a prueba estos conocimientos recibidos en una experiencia real, pero no cuenta con la disponibilidad para ir a un yacimiento arqueológico y realizar una excavación. Por estas razones, Gina decide hacer eso del simulador de excavaciones arqueológicas desarrollado en el presente proyecto de tesis.</p> <p>Gina ingresará al simulador de excavaciones arqueológicas e iniciará una nueva simulación. En ella, Gina podrá seleccionar y visualizar el tipo de prospección arqueológica que ella decida. Además, podrá experimentar asignando diferentes tipos</p>

de trabajadores a cada yacimiento, visualizando el detalle de la información de cada trabajador. Asimismo, al pasar a la etapa de excavación arqueológica, podrá ver el efecto de utilizar una herramienta u otra y su impacto en la excavación de artefactos. Después, podrá visualizar la información que se registra para cada artefacto encontrado. Finalmente, cuando termine de excavar todos los artefactos posibles en los yacimientos que prospectó, finalizará la simulación y volverá al menú principal.

4.1.4 Formulación de historias de usuario

A partir de la entrevista con la Dra. Fernandini y la experiencia al utilizar el simulador “El Pez Dorado”, se obtuvo la lista de catálogo de requisitos. Esta lista puede encontrarse en el anexo F del presente proyecto.

4.2 Validación académica de los aspectos a ser incluidos en el simulador de excavaciones arqueológicas

De acuerdo a la metodología que se sigue en el presente proyecto, una combinación de *Extreme Programming* y *User-Centered Design*, se realizó una validación académica de las historias de usuario formuladas en el punto 4.1. Esta validación se llevó a cabo a través de la presentación de prototipos de diecinueve (19) pantallas del simulador a desarrollarse a la experta en la docencia escogida. En primer lugar, se realizaron prototipos en papel de cómo se deberían ver las pantallas del simulador. Ejemplo de las pantallas prototipadas en papel se muestran a continuación.

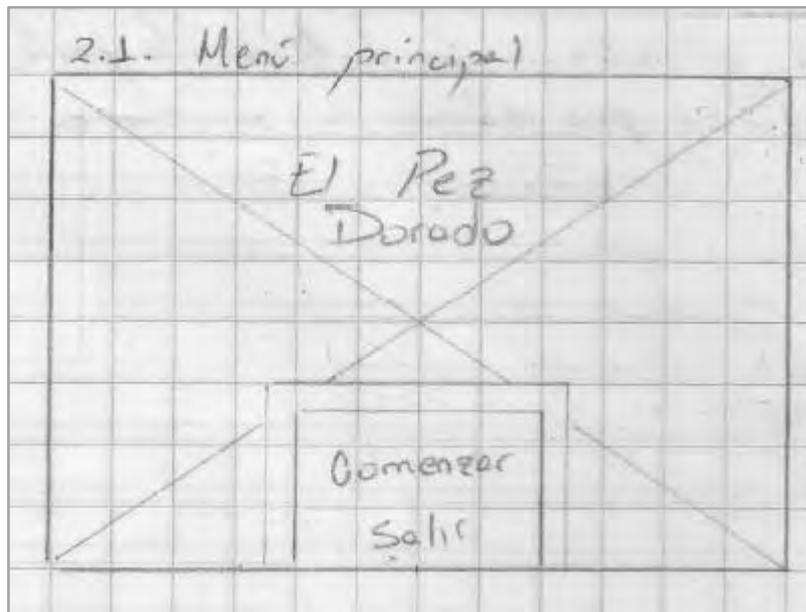


Figura 3. Prototipo en papel de pantalla de menú principal (Elaboración propia)



Figura 4. Prototipo en papel de pantalla de prospección arqueológica (Elaboración propia)

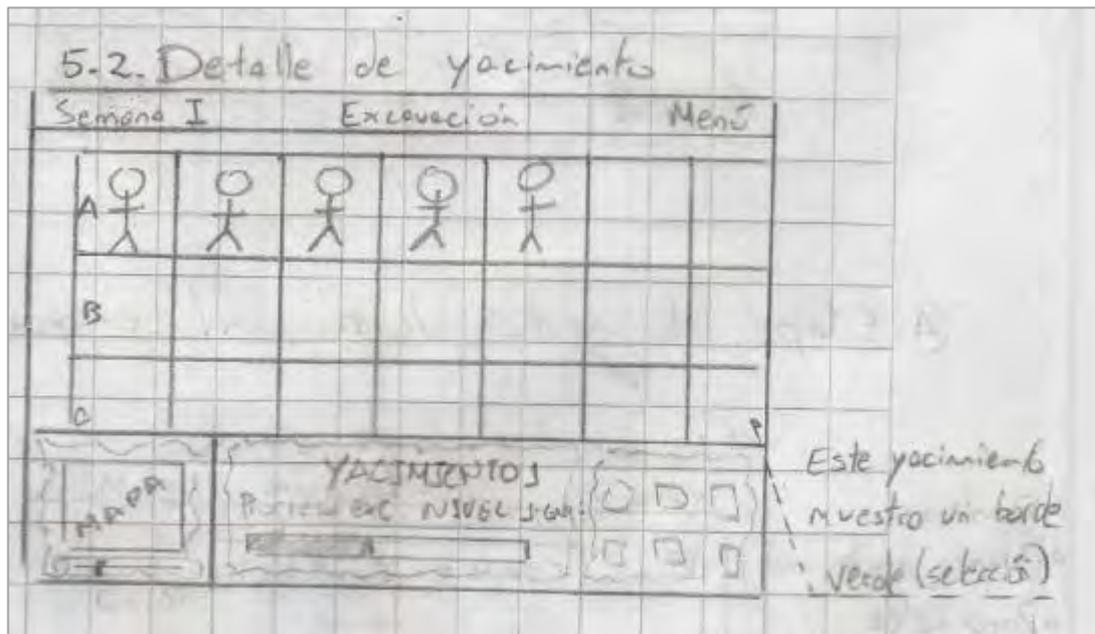


Figura 5. Prototipo en papel de pantalla de excavación arqueológica (Elaboración propia).

Posteriormente, una vez concluido todo el prototipado en papel, se construyó una versión digital de estos prototipos con los recursos del simulador “El Pez Dorado”, de manera que estos puedan ser presentados a la experta en la docencia. A continuación, se muestran algunos prototipos que se formularon.



Figura 6. Prototipo de pantalla de menú principal (Elaboración propia)



Figura 7. Prototipo de pantalla de detalle de método de muestreo (Elaboración propia)

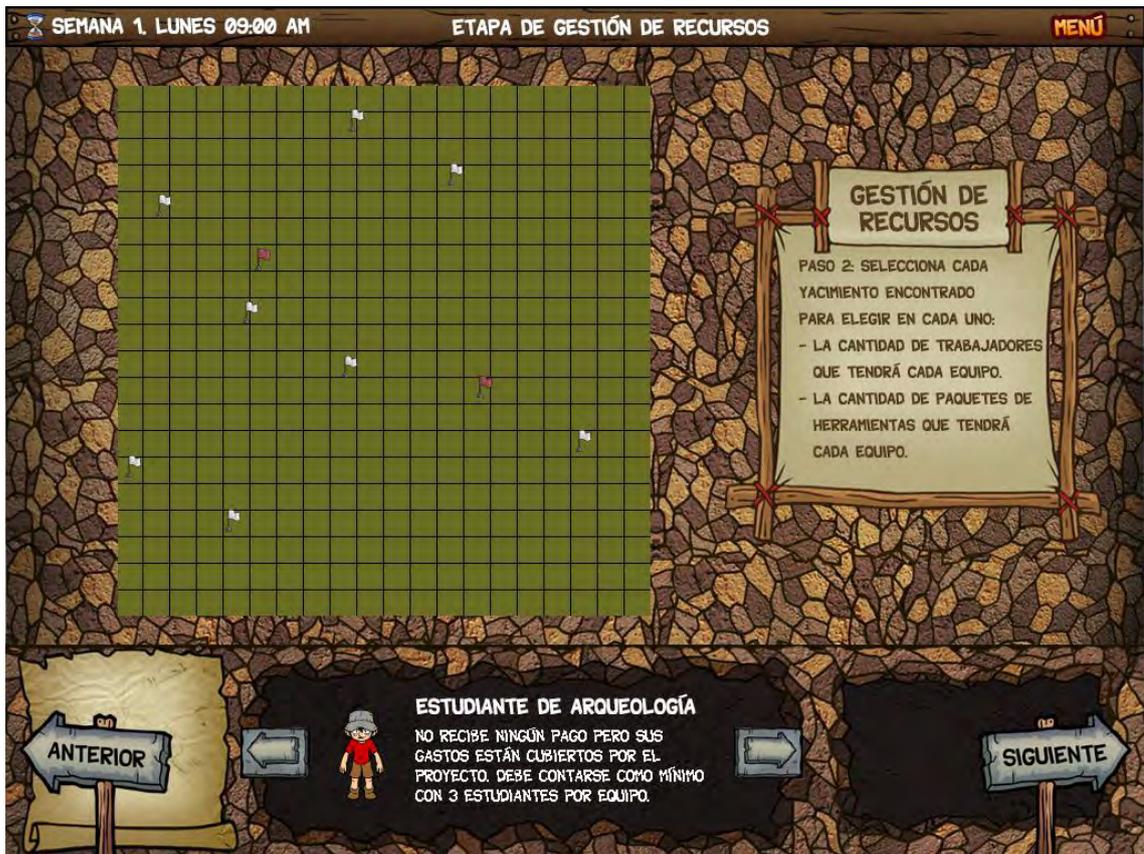


Figura 8. Prototipo de la pantalla de gestión de recursos (Elaboración propia)





Figura 9. Prototipo de la pantalla de excavación (Elaboración propia)



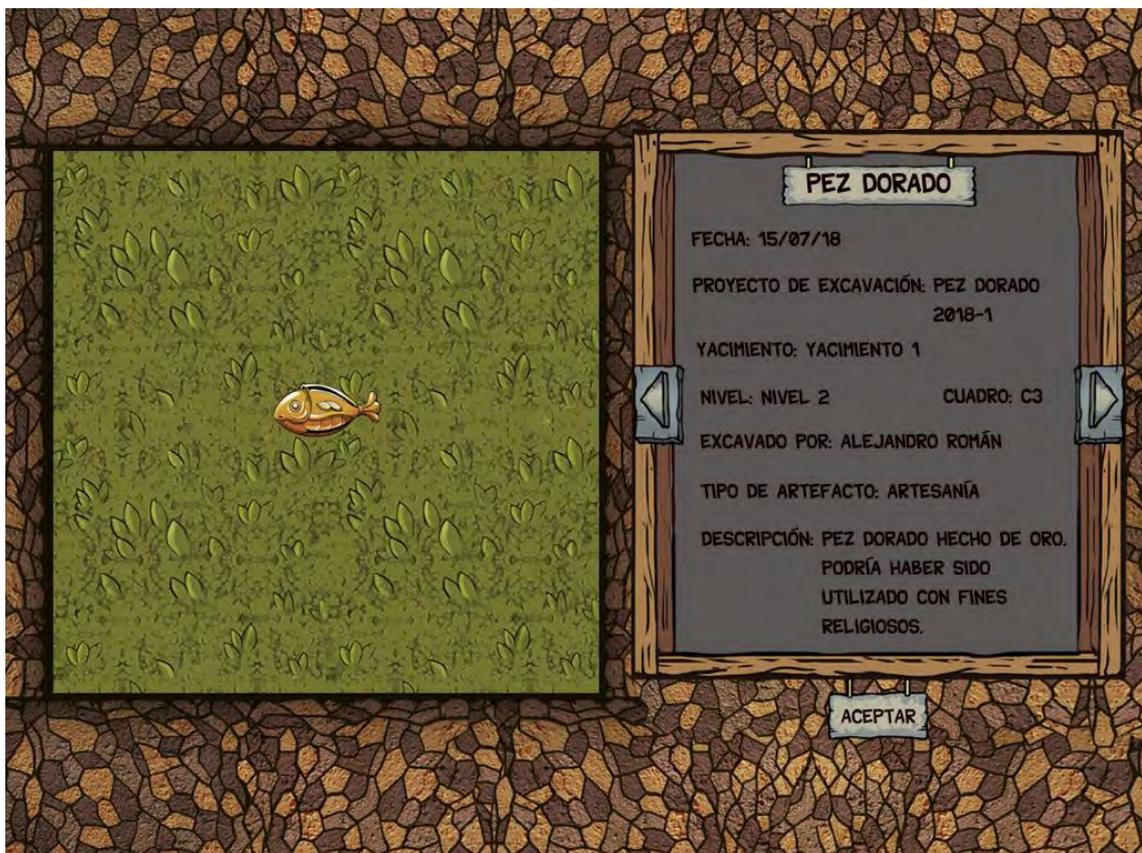


Figura 10. Prototipo de inventario de artefactos encontrados (Elaboración propia)

En el anexo H pueden encontrarse el resto de pantallas prototipadas, tanto en papel como de manera digital.

Posteriormente, se tuvo una reunión con la experta en la docencia de arqueología, la doctora Francesca Fernandini. En esta reunión, se le explicó el flujo general del simulador, sus funcionalidades, y se realizó una simulación de cómo actuaría el simulador, a partir de las pantallas prototipadas en digital. Finalmente, al concluir la explicación del simulador, se le brindó a la experta en la docencia un cuestionario sobre el cumplimiento académico de las funcionalidades del simulador, el cual puede ser encontrado en el anexo I. En este cuestionario, se incluyeron cinco enunciados sobre el cumplimiento académico del simulador distribuidos de la siguiente manera: Un enunciado sobre el simulador en general, un enunciado sobre la etapa de prospección arqueológica, un enunciado sobre la etapa de gestión de recursos, un enunciado sobre la etapa de excavación arqueológica y un enunciado de conclusión. Cada uno de estos

enunciados podía ser respondido con “Sí” o “No” respecto a si se siguen los principios académicos de la arqueología. Además, se brindó un campo “Observaciones” para que la experta pueda brindar mayor detalle. De los cinco enunciados, cuatro fueron contestados afirmativamente, incluyendo el enunciado de conclusiones. El único enunciado que no pasó la evaluación es el relacionado con la etapa de excavación. Esto se debió a la inconformidad de la experta en docencia respecto a la posibilidad de destrucción de restos arqueológicos. Dado que destruir restos arqueológicos es un delito, indicó que no se puede agregar ese mensaje en un simulador que siga los principios académicos de la arqueología. En su lugar, se sugiere mencionar que se podría afectar el yacimiento arqueológico o al artefacto encontrado, pero no mencionar su destrucción.

A modo de resumen, dado que se contestó afirmativamente en la mayoría de los casos, incluyendo la sección de conclusiones, se afirma que las funcionalidades cubiertas por el presente proyecto sí cumple con los principios académicos de la arqueología.

Capítulo 5. Diseño de la arquitectura de software que soporte la simulación de excavaciones arqueológicas

En el presente capítulo, se diseñará, como parte del resultado esperado 3, una arquitectura de software que soporte la simulación de excavaciones arqueológicas. Además, se validará la característica de rendimiento de esta arquitectura de software a través del método “Software Architecture Analysis Method” o “Método de análisis de arquitecturas de software” descrito en el capítulo 1. Posteriormente, como parte del resultado esperado 4, se someterá al software desarrollado en los objetivos específicos 3 y 4 a una prueba de estrés donde se evaluará si este software desarrollado tiene o no un mejor consumo promedio de recursos computacionales que el simulador inestable que se tenía anteriormente.

De esta manera, el presente capítulo se divide en las siguientes partes:

1. Diseño de la arquitectura de software que soporte la simulación
2. Validación del rendimiento de la arquitectura planteada a través del método “Software Architecture Analysis Method”
3. Prueba de estrés comparativa con el simulador “El Pez Dorado” original.

5.1 Diseño de la arquitectura de software que soporte la simulación

A partir de las funcionalidades identificadas en el capítulo anterior, se diseñó la arquitectura de software que deberá tener el simulador a desarrollarse en los objetivos específicos 3 y 4. Dado a que el software a diseñarse busca tener un menor consumo promedio de recursos computacionales que el simulador “El Pez Dorado” anterior, se buscó bibliografía que describa patrones para sistemas de alto rendimiento y alta interactividad. De esta manera, se leyeron los libros “*Game programming patterns*” (Nystrom, 2014) y “*SFML Game development*” (Moreira, Haller, & Hansson, 2013) para obtener patrones de diseño de referencia.

De esta manera, se describen a continuación los diagramas con los que se representó a la arquitectura de software. Estos diagramas corresponden a las vistas descritas por el modelo de vista de arquitectura 4 + 1. Se escogió este modelo de vista ya que considera en sus representaciones los requerimientos no funcionales de rendimiento (Kruntschen, 1995, p. 4,5). Sin embargo, es necesario resaltar que, como se indicó

anteriormente, no se utilizará la vista de escenarios, sino que se utilizarán las historias de usuario identificadas en el capítulo 4 en su lugar. Estos diagramas pueden ser encontrados en el anexo G.

5.1.1 Vista lógica: Diagrama de clases

La primera vista de la arquitectura que se presenta es la vista lógica, a través de un diagrama de clases. Este diagrama muestra las clases que se utilizarán para la representación lógica de las entidades del simulador.

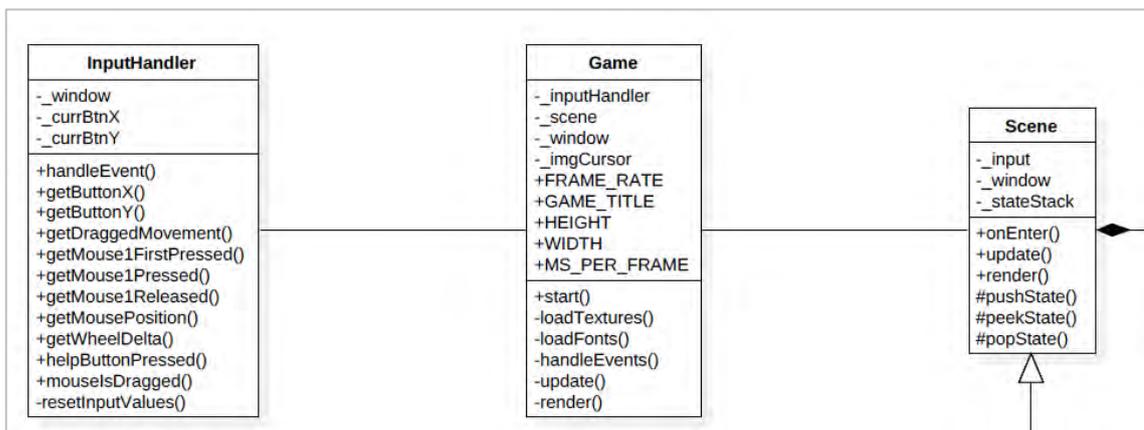


Figura 11. Clase "Game" (Elaboración propia, Anexo G)

El componente principal de este diagrama es la clase "Game", el cual se muestra en la figura 11. Es necesario resaltar que, a pesar de que no se trata de una experiencia lúdica, se escogió este nombre debido a los patrones seguidos según la bibliografía descrita anteriormente. Dado que el presente desarrollo busca disminuir el uso de recursos computacionales a comparación del simulador previo, se ha utilizado un patrón que no depende de la ejecución de múltiples hilos para realizar la simulación. Este patrón consiste iterar, en un solo hilo, un bucle donde se realizan tres acciones:

1. Manejar eventos de entrada o salida ("Input" y "Ouput")
2. Actualizar el estado interno de los elementos del simulador
3. Dibujar los elementos actualizados del simulador

La primera acción, llevada a cabo en la función "handleEvents" de la clase "Game", se encarga de guardar, mediante la clase "InputHandler", todos los eventos del teclado y del ratón que detecta el sistema operativo. Luego, la segunda acción, llevada a cabo en la función "update", actualizará el estado interno de todos los elementos del simulador a partir de los eventos registrados. Finalmente, tras realizar la actualización de los

elementos del simulador, se culmina el bucle con el dibujado de los elementos del simulador en pantalla. Esta acción se realiza a través de la función *render*. Por esta razón, que la clase “*Game*” agrupa el bucle principal del simulador, es que se considera como la clase principal de la arquitectura.

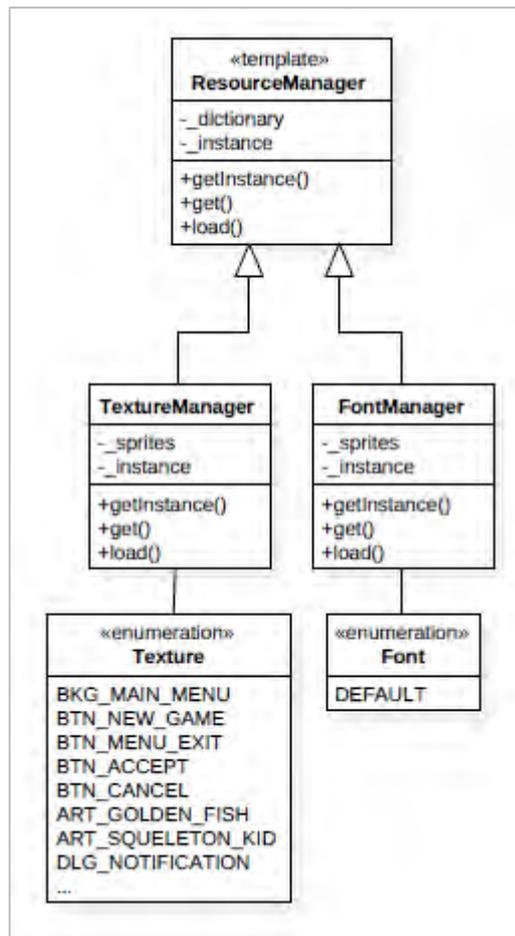


Figura 12. Clase “*ResourceManager*” (Elaboración propia, Anexo G)

Sin embargo, a pesar de ser el componente principal, la implementación de las funcionalidades de la clase “*Game*” es delegada a otras clases. Como ya se mencionó, el manejo de la interacción con los eventos de entrada y salida del sistema es manejado a través de la clase “*InputHandler*”. Asimismo, la carga, almacenamiento y acceso a todos los recursos externos, como imágenes o fuentes, que se requieren utilizar en el simulador son delegados a la clase plantilla “*ResourceManager*”. Esta clase se especializa en “*TextureManager*” y “*FontManager*” para manejar texturas y fuentes, respectivamente.

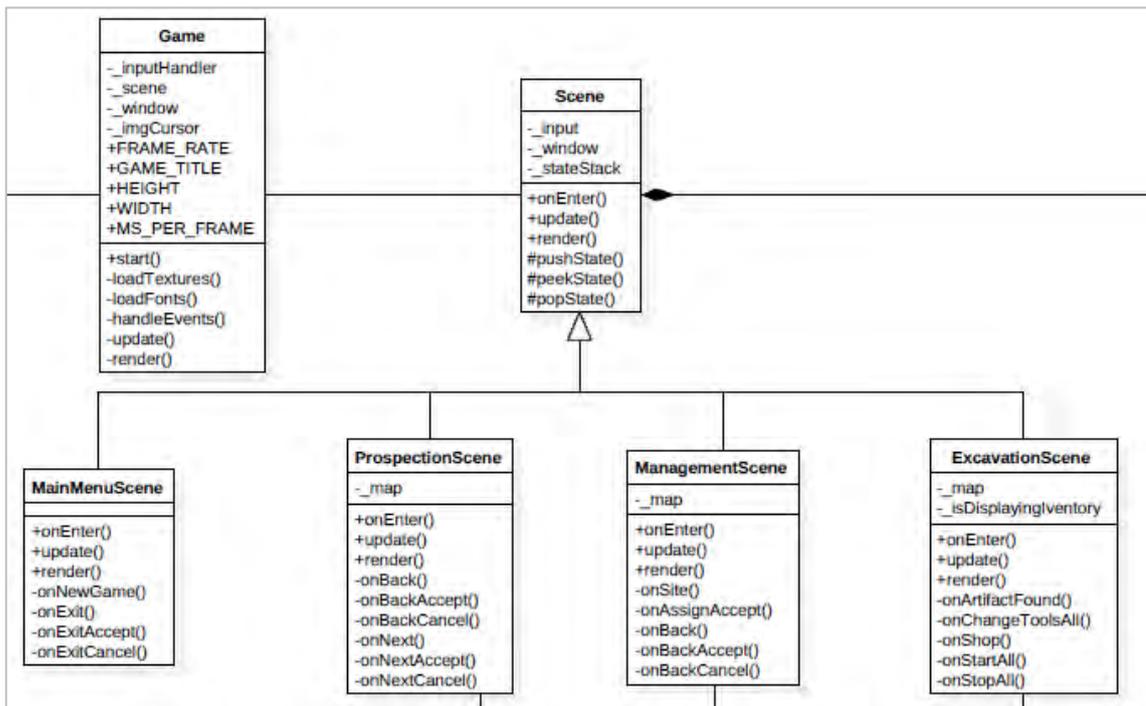


Figura 13. Clase "Scene" (Elaboración propia, Anexo G).

Así también, la actualización y dibujo de los elementos internos del simulador es delegado a las clases "Scene". Estas clases representan las etapas que componen el flujo de la simulación: Menú principal, etapa de prospección arqueológica, etapa de gestión de recursos y etapa de excavación arqueológica. Cada escena contiene todos los elementos necesarios para representar la etapa que le corresponde y se encarga de actualizarlos y dibujarlos. De esta manera, cada escena encapsula los elementos de la etapa que les corresponde. Es necesario resaltar que la clase "Game" contiene una variable miembro "_scene". Esta variable miembro representa la escena actual que se está mostrando. Así, solo se tiene en memoria y se procesa una escena a la vez, por lo que se evita estar almacenando y procesando elementos que no corresponden a la etapa actual. De esta forma, se disminuye el consumo de memoria RAM y consumo de CPU.

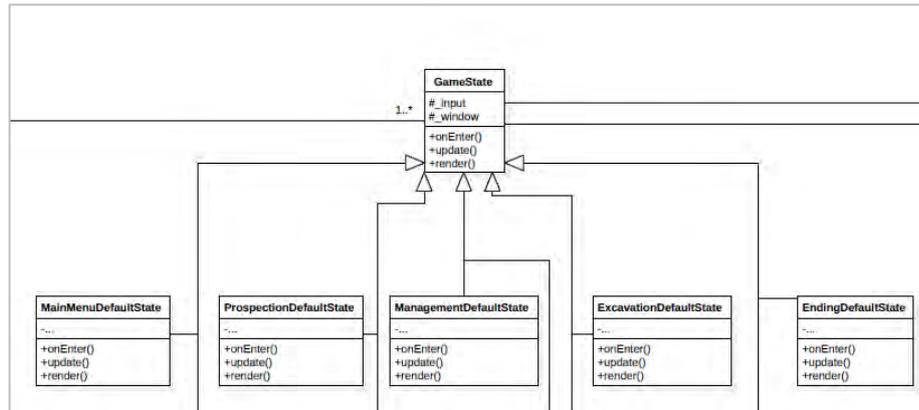


Figura 14. Clase "GameState" (Elaboración propia, Anexo G).

Adicionalmente, el almacenamiento y procesamiento de los elementos del simulador de cada escena se delegan a las clases derivadas de la clase "GameState". Cada una de estas clases representan un estado del simulador dentro de la escena. Así, se tienen estados en los que se muestran mensajes de confirmación (clase "NotificacionState") o estados para comprar herramientas de excavación (clase "BuyResourcesState"). Al igual que con la clase "Scene", esta división permite solo procesar los elementos del estado actual y no otros elementos inactivos, ahorrando los recursos computacionales mencionados. Además, el uso de estas clases permite reutilizar el código de una clase (como la clase "ConfirmationState") en diferentes escenas.



5.1.2 Vista física: Diagrama de despliegue

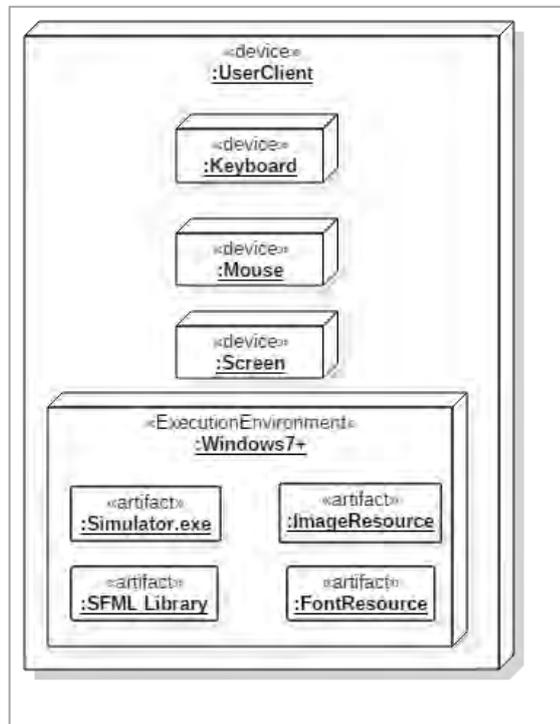


Figura 15. Vista física: Diagrama de despliegue (Elaboración propia, Anexo G)

La segunda vista utilizada es la vista física, a través de un diagrama de despliegue. En este diagrama se muestra los entornos, tanto físicos como digitales, que son necesarios para poder utilizar el simulador. De esta manera, se reconoce que se necesitan dos entornos principales para ejecutar el software desarrollado: un computador que cuente con teclado, cursor y pantalla y un sistema operativo que sea Windows XP o superior, Linux o macOS. Este último requerimiento se debe a los requerimientos mínimos que soporta la librería gráfica “SFML” que fue utilizada.

Asimismo, también se han identificado artefactos adicionales necesarios para la ejecución del simulador. Estos son: la librería gráfica “SFML”; imágenes, las cuales van a ser dibujadas en pantalla; y fuentes, las cuales se utilizarán para mostrar texto.

5.1.3 Vista de proceso: Diagrama de actividades

La tercera vista utilizada fue la vista de proceso, a través de diagramas de actividades. Dado que esta vista explica los procesos del software, se han creado cuatro diagramas,

uno por cada uno de los procesos principales del simulador.

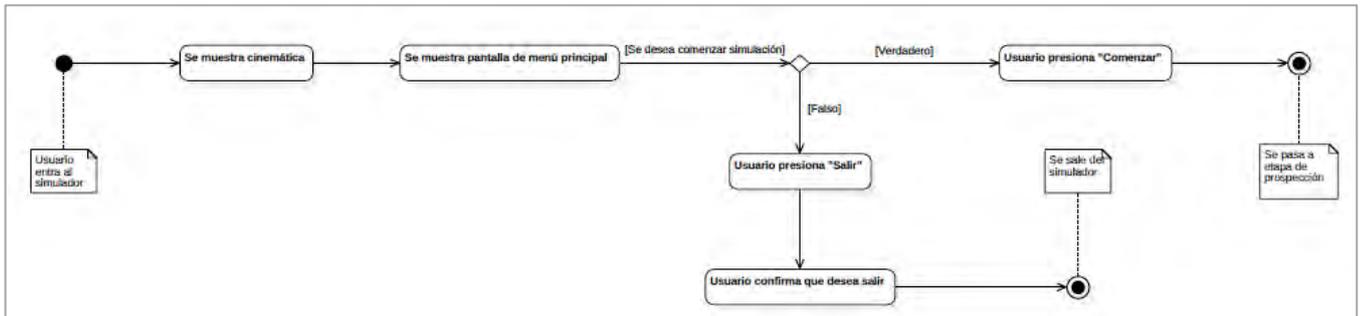


Figura 16. Vista de proceso: Diagrama de actividades - Proceso de menú principal
(Elaboración propia, Anexo G)

En primer lugar, se explicará el diagrama de actividades del proceso de menú principal. Este diagrama muestra el flujo realizado por el usuario al ingresar al simulador e interactuar con el menú principal. En un inicio, se muestra una cinemática, para luego mostrarse dos opciones: “Comenzar” y “Salir”. Si el usuario presiona “Comenzar”, se pasará a la etapa de prospección arqueológica y, si el usuario presiona “Salir”, se mostrará un mensaje de confirmación antes de salir del simulador.

En segundo lugar, se diagramó el proceso de prospección arqueológica. En este flujo, el usuario recibe información sobre la etapa, selecciona un método de muestreo para realizar la prospección, visualiza la información sobre el método de muestreo, realiza la prospección y recibe información sobre la prospección realizada. Finalmente, pasa a la siguiente etapa: Gestión de recursos.

En tercer lugar, se diagramó el proceso de gestión de recursos. Este diagrama, al igual que el anterior, inicia mostrando información al usuario sobre la etapa a desarrollarse. Sin embargo, se diferencia de la etapa anterior en que permite mayor configuración al usuario. Así, tras haber recibido información sobre cada tipo de recurso a utilizar, el usuario debe seleccionar cada yacimiento encontrado durante la etapa de prospección y asignarle recursos humanos y materiales. Finalmente, tras haber asignado recursos a todos los yacimientos, se pasa a la etapa de excavación arqueológica.

En cuarto lugar, se diagramaron las actividades del proceso de excavación. Este diagrama es el que más actividades presenta de los cuatro diagramas de actividades presentados. Esto se debe a que contiene múltiples opciones de configuración para realizar la excavación.

El proceso comienza, al igual que los dos procesos anteriores, con una explicación de

la etapa a desarrollarse. Luego, permite al usuario asignar herramientas de excavación a los trabajadores de uno o todos los yacimientos. A continuación, se brinda la opción de iniciar la excavación en uno o todos los yacimientos, a criterio del usuario. Es necesario resaltar que se incrementa la complejidad del diagrama debido a que cada una de estas dos opciones se pueden realizar múltiples veces a lo largo del proceso, por lo que se inserta un bucle y un condicional para cada una de ellas. Después, continuando con el flujo, se realiza una verificación si es que se ha encontrado un artefacto o no. De ser negativa la respuesta, se vuelve a iterar en las dos opciones mencionadas. De ser positiva la respuesta, se muestra un recuadro de notificación y se da la opción al usuario de cambiar las herramientas de excavación para evitar que se afecte al artefacto descubierto. Si no se utilizan las herramientas correctas, se notifica al usuario de que el artefacto ha sido destruido. Caso contrario, se notifica al usuario de que el artefacto ha sido recuperado con éxito para luego mostrar el inventario con todos los artefactos encontrados. En ambos casos, se evalúa si se ha finalizado una semana de excavación o no. De ser así, se muestra una ventana de resumen sobre los artefactos descubiertos en la semana. Caso contrario, se vuelve a iterar todo el proceso mencionado.

Finalmente, si ha finalizado el tiempo de excavación, se pasa a la pantalla de fin de juego. En esta pantalla, el usuario solo podrá presionar “Volver a menú principal” para regresar al menú principal y, así, concluir el proceso de excavación arqueológica.

5.1.4 Vista de desarrollo: Diagrama de componentes

La cuarta y última vista utilizada fue la vista de desarrollo, a través de un diagrama de componente. En este diagrama se muestran los principales componentes utilizados en el desarrollo del simulador planteado en el presente proyecto. De manera general, este diagrama explica las interacciones que se tienen entre las principales clases detalladas en el diagrama de clases.

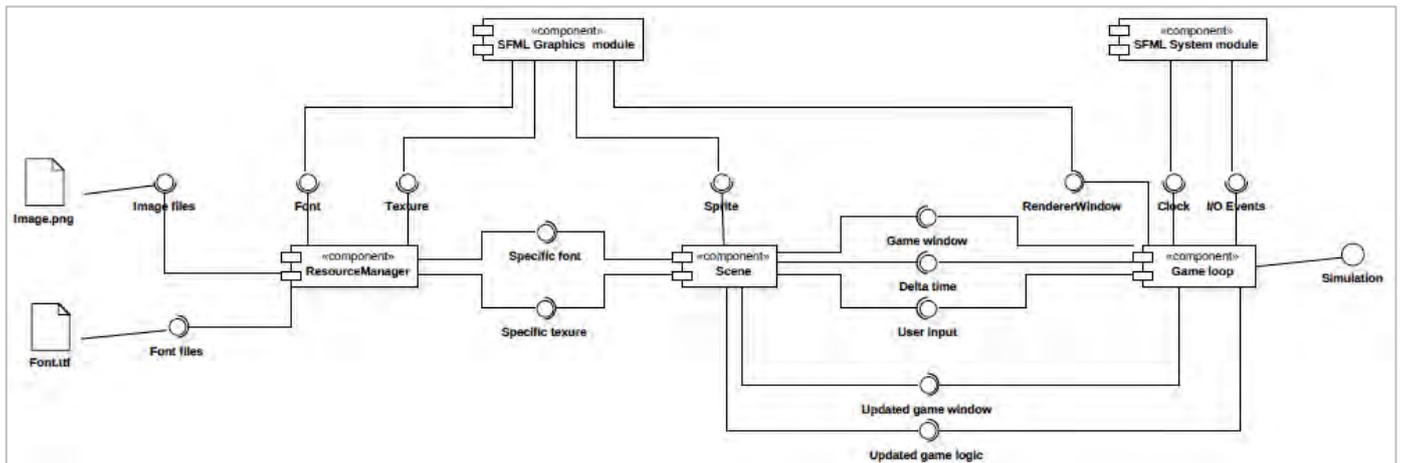


Figura 17. Vista de desarrollo: Diagrama de componentes (Elaboración propia, Anexo G)

El componente principal del diagrama es el componente “*Game loop*”. Este componente se encarga de brindar, a alto nivel, la simulación propiamente dicha. Esto lo logra a la iteración de un bucle que contiene tres etapas. En primer lugar, maneja los eventos I/O y el tiempo transcurrido, mediante interacción con el módulo “*System*” de la librería SFML. En segundo lugar, brinda la ventana “*Game window*” en la cual se debe dibujar los elementos de la simulación, el tiempo transcurrido “*delta time*” desde la última iteración y el “*user input*” o entrada del usuario. Finalmente, recibe la lógica actualizada de la simulación y la ventana del simulador con los gráficos actualizados. Si se compara con el diagrama de clases, se encuentra su equivalente en la clase “*Game*”.

En segundo lugar, otro componente principal es el componente “*Scene*”. Este componente agrupa el conjunto de clases “**Scene*” en el diagrama de clases. Su principal función es actualizar la lógica de la simulación, según el estado que se tenga, y actualizar los elementos dibujados en la ventana de la simulación. Para ello, requiere de la clase “*Sprite*” del módulo “*Graphics*” de la librería gráfica “*SFML*”. Asimismo, también requiere de fuentes y texturas en específicos a través del componente “*ResourceManager*”.

Finalmente, se tiene como tercer componente principal al componente “*ResourceManager*”. Este componente tiene como finalidad proveer de texturas y fuentes al componente “*Scene*”. Es decir, provee de recursos a las demás clases “*Scene*”. Para ello, requiere tanto las clases “*Font*” y “*Texture*” del módulo “*Graphics*” de

la librería “SFML” como de los archivos de imágenes, en formato PNG, y los archivos de fuentes, en formato TTF.

5.2 Validación del rendimiento de la arquitectura planteada a través del método “Software Architecture Analysis Method”

En esta sección se validará la característica de rendimiento de la arquitectura de software diseñada en la sección anterior. Para ello, se utilizará, como se mencionó en el capítulo 1, el método “Software Architecture Analysis Method” o “Método de análisis de arquitectura de software”. Las actividades que componen el método ya han sido descritas en el capítulo 1. Sin embargo, no se realizarán todas las actividades que componen el método. Esto se debe a que la cuarta actividad mide el grado en el que la arquitectura considera una separación de intereses de sus módulos (Kazman et al., 1996, p. 50). Dado que la característica de interés solo es el rendimiento del software a desarrollarse y no el acoplamiento de sus módulos, no se considera necesario realizar esta actividad. Asimismo, dado que la quinta actividad consiste en ordenar múltiples arquitecturas a comparar según la característica de interés, no se ha considerado necesario realizarla en este caso en el que solo se tiene una única arquitectura a evaluar. De esta manera, se realizarán solo las tres primeras actividades, como ha sido realizado el método en otro trabajo (Zbick, 2017).

De esta manera, el método a seguir se compone de las siguientes tres actividades:

1. Descripción de la arquitectura candidata

En esta actividad se describe la arquitectura a evaluar según la característica de rendimiento, de manera que todos los interesados puedan entenderlo de manera uniforme. Sin embargo, dado que ya se ha descrito la arquitectura en la sección anterior, se omitirá esta actividad.

2. Desarrollo de escenarios

El método “Software Architecture Analysis Method” es un método basado en escenarios (Zbick, 2017, p. 234). De esta manera, se formularán escenarios que deberá soportar la arquitectura diseñada. Para el presente proyecto, se han considerado escenarios relacionadas a la característica evaluada: rendimiento.

3. Evaluación de cada escenario

Para cada escenario identificado en la actividad anterior, se deberá evaluar si la arquitectura evaluada puede soportarlo actualmente o requiere algún cambio adicional para hacerlo. Para los fines de este proyecto, solo considerará que, si se requiere un cambio para soportar el escenario, la arquitectura planteada no cumple con el atributo de rendimiento.

De esta manera, se presentan las actividades 2 (elección de escenarios) y 3 (evaluación de escenarios) en la siguiente tabla:

Tabla 12. Actividades 2 y 3 del método SAAM (Elaboración propia)

Escenario	Motivo de elección	Evaluación de rendimiento
El usuario comienza la ejecución del software al ingresar al simulador.	Este es el escenario inicial a través del cual se accede a la simulación. El software debe ser capaz de soportarlo.	Dado que solamente se muestran los recursos necesarios en el menú principal y a que el simulador pausa el hilo que ejecuta el proceso hasta que se pase a otra etapa, se considera que este escenario cumple la característica de rendimiento y no requiere cambios adicionales.
El simulador recibe la interacción del usuario con los dispositivos de entrada teclado y cursor.	A lo largo de la simulación, el usuario interactuará con el software mediante el teclado y el cursor. Por ello, el simulador debe ser capaz de soportar este escenario.	Dado a que se almacena las interacciones con el usuario en la clase "InputHandler" de manera periódica, según el componente "GameLoop", y a que no se bloquea la ejecución del simulador en espera de mayor interacción, sino que continúa la ejecución, se considera que la arquitectura cumple la característica de rendimiento en este escenario.
El usuario interactúa con un botón del	El usuario deberá presionar múltiples botones a lo largo de la simulación	Dado que solo se verifica la interacción con los botones del

<p>simulador.</p>	<p>para poder pasar a las siguientes etapas del proyecto de excavación arqueológica. Por ello, el simulador debe ser capaz de soportar este escenario.</p>	<p>simulador cuando el usuario presiona el cursor, evitando iterar por todos los botones para verificar la interacción en cada bucle de "GameLoop", se considera que la arquitectura planteada cumple la característica de rendimiento en este escenario.</p>
<p>El simulador muestra la etapa de menú principal, prospección arqueológica, gestión de recursos, excavación arqueológica o fin de juego.</p>	<p>Para que el usuario se informe acerca de un proyecto de excavación arqueológica, el simulador debe poder ser capaz de mostrar las distintas etapas que lo componen.</p>	<p>Dado que solo se muestran los gráficos de una etapa a la vez, solo se realiza procesamiento de data relacionada a esa etapa. Las actividades se realizan de manera discreta en un bucle y no de manera continua a través de hilos. Por ello, se considera que la arquitectura planteada cumple la característica de rendimiento en este escenario.</p>
<p>El simulador muestra la etapa de excavación con un solo yacimiento en pantalla y con la mínima cantidad de trabajadores.</p>	<p>Durante la etapa de excavación, el simulador puede mostrar uno o muchos yacimientos en pantalla. Por ello, el software debe poder ser capaz de mostrar la configuración mínima (un solo yacimiento con la cantidad mínima de trabajadores).</p>	<p>Dado que el simulador dibuja únicamente el yacimiento en pantalla y ningún otro adicional y a que cada trabajador se dibuja de manera discreta desde un único hilo y no de manera continua a través de múltiples hilos, se considera que la arquitectura planteada cumple la característica de rendimiento.</p>
<p>El simulador muestra la etapa de excavación con todos los yacimientos en pantalla y con la máxima cantidad de</p>	<p>Durante la etapa de excavación, el simulador puede mostrar uno o muchos yacimientos en pantalla. Por ello, el software debe poder ser capaz de mostrar la configuración máxima (todos los yacimientos con la cantidad</p>	<p>Dado que el simulador dibuja cada elemento de manera discreta desde un único hilo, a través del bucle descrito en "GameLoop", y no de manera continua a través de múltiples hilos, se considera que la</p>

trabajadores en cada yacimiento excavado.	máxima de trabajadores).	arquitectura planteada cumple la característica de rendimiento.
El simulador realiza la carga de imágenes y fuentes.	Dado que el simulador muestra en pantalla múltiples recursos visuales, este debe poder soportar la carga estos recursos.	Dado que los recursos externos son almacenados en la clase “ResourceManager” y que esta provee los recursos según se necesiten, no es necesario realizar la carga de un mismo recurso múltiples veces. Por ello, se considera que la arquitectura planteada cumple la característica de rendimiento.

Debido a que, tras realizar la evaluación, ningún escenario requiere modificaciones para soportar los escenarios descritos relacionados a rendimiento, se considera que la arquitectura de software planteada cumple la característica de rendimiento.

5.3 Prueba de estrés comparativa con el simulador “El Pez Dorado” original

En esta sección, se detallarán las acciones realizadas para alcanzar el resultado esperado 4 “Resultados de pruebas de estrés del simulador desarrollado con la arquitectura diseñada y que demuestre un mejor rendimiento que el simulador ‘El Pez Dorado’”.

En primer lugar, se hizo necesario definir qué características serían las que se medirían para que se demuestre que el simulador desarrollado tuvo un mejor rendimiento que el simulador original. Con este objetivo, se escogieron características a comparar en base a un trabajo de comparación de rendimiento de simuladores de redes inalámbricas (Ur Rehman Khan et al., 2013). En este trabajo, se realizó la comparación de rendimiento basado en cuatro características: uso de memoria (MB), utilización del CPU (porcentaje), escalabilidad y tiempo de computación (segundos) (Ur Rehman Khan et al., 2013, p. 37). Sin embargo, no se consideró como característica aplicable a este proyecto la característica de tiempo de computación. Esto se debe a que se consideraba como tiempo de ejecución al tiempo en que se demoraba el simulador en ejecutar un

algoritmo de ruteo determinado. Dado que el presente simulador no cuenta con un algoritmo en específico para la determinación de soluciones óptimas, se ha considerado esta característica como no aplicable. Por otra parte, según el testimonio oral de Luis Robles (Ver anexo B) y de la revisión del código fuente que realizó el autor del presente documento, se determinó que se hacía un uso de hilos innecesario. Por esta razón, para comparar un posible ahorro de este recurso computacional, se ha tenido en cuenta este recurso computacional al momento de comparar la ejecución del simulador desarrollado y el simulador original.

Tras haber definido las características a medirse, se procedió a definir los parámetros con los cuales se realizarían las mediciones. Estos parámetros también se basaron en la comparación de simuladores anteriormente mencionada (Ur Rehman Khan et al., 2013, p. 37). A continuación, se presenta la tabla de resumen de los parámetros.

Tabla 13. *Parámetros de prueba de rendimiento*

Parámetro	Valor
Tiempo de medición por configuración	60 segundos
Número de configuraciones	7
Número de trabajadores	60 – 80
Intervalos de medición	1 segundo

Las configuraciones sobre las cuales se aplicarán estos parámetros son 7: una para la configuración que se muestra durante la etapa de prospección, una para la etapa de gestión de recursos y cinco para la etapa de excavación. Respecto a la etapa de excavación, en cada configuración se incrementará de manera uniforme el número de trabajadores en pantalla. A diferencia de la comparación de rendimiento de referencia, se ha reducido el tiempo de medición por configuración. Esto se debe a que en el trabajo de referencia la medición incluye el tiempo de ejecución del algoritmo de ruteo. Dado que el presente trabajo no contempla la ejecución de un algoritmo similar, se redujo el tiempo de medición que se invertiría en la ejecución del algoritmo. Asimismo, el tiempo

de intervalos de medición se incrementó de 0.2 segundos a 1 segundo. Esto se debe a que la medición en el trabajo de referencia se debía a la configuración del algoritmo que se utilizaba. De igual manera, dado que se cuenta con un algoritmo similar, en el presente trabajo se tomó como intervalo de medición el tiempo mínimo de medición de la herramienta utilizada.

Por otra parte, es necesario resaltar que los simuladores presentan variaciones entre la ejecución de sus distintas etapas (prospección, gestión y excavación). Por ejemplo, durante la prospección del simulador desarrollado se simula la pérdida de yacimientos seleccionados debido a que no contenían restos arqueológicos, mientras que en el simulador original no este efecto no se simula. Esto se debe a que el simulador desarrollado consideró la validación de los aspectos académicos incluidos en los requerimientos del simulador, por lo que puede incluir características diferentes al simulador original. Por esta razón, para realizar una simulación en configuraciones similares, se realizaron cambios al código fuente del simulador desarrollado para que la configuración del simulador desarrollado sea lo más cercana a la configuración del simulador original.

De esta manera, se realizó la medición de cada configuración de cada simulador. La medición se realizó a través de la herramienta "Performance Monitor" del sistema operativo Windows. Respecto a la máquina donde se desarrollaron las pruebas, las especificaciones se han detallado anteriormente en la sección de 1.3.6. Los resultados de las mediciones pueden ser encontrados en el anexo K.

Posteriormente, para determinar qué simulador ha tenido un mejor desempeño en las características indicadas, se hizo necesario realizar una experimentación numérica sobre las mediciones realizadas. De esta manera, se identificaron las dos siguientes actividades a realizarse:

1. Determinación de la normalidad de los conjuntos de datos
2. Definición y comprobación de hipótesis sobre las características de rendimiento

A continuación, se describe cada una a detalle.

5.3.1 Determinación de la normalidad de los conjuntos de datos

Esta actividad fue necesaria ya que las hipótesis que se realicen dependerán del tipo de conjunto de datos que se tiene. Por ejemplo, si se trata de un conjunto de datos que siguen una distribución normal, se podrían realizar y comprobar hipótesis sobre sus

medias. Por el contrario, si se trata de un conjunto de datos que no siguen una distribución normal, se podrían realizar y comprobar hipótesis sobre sus medianas.

De esta manera, para determinar la normalidad del conjunto de datos, se realizó la prueba de normalidad Shapiro-Wilk (Shapiro & Wilk, 1965). En esta prueba, se tienen como hipótesis nula y alternativa (Shapiro & Wilk, 1965, p. 602):

H_0 : El conjunto de datos proviene de una distribución normal

H_1 : El conjunto de datos no proviene de una distribución normal

A partir de estas hipótesis, se realizó la evaluación con un nivel de significancia del 95%. Los resultados de las pruebas, a través del valor p , se resumen en la siguiente figura.

		Número de hilos	% Procesador	Memoria virtual (Bytes)
Etapa de prospección	Antiguo	8.70E-12	1.59E-05	8.70E-12
	Nuevo	8.17E-09	3.39E-02	8.17E-09
Etapa de gestión	Antiguo	8.49E-13	3.71E-04	8.49E-13
	Nuevo	-	1.16E-02	1.45E-14
Etapa de excavación - 60 trabajadores	Antiguo	8.08E-06	2.97E-02	6.54E-06
	Nuevo	-	3.40E-02	1.83E-07
Etapa de excavación - 65 trabajadores	Antiguo	5.37E-07	3.03E-04	4.98E-07
	Nuevo	9.57E-12	1.20E-02	1.12E-09
Etapa de excavación - 70 trabajadores	Antiguo	3.60E-07	2.59E-06	3.72E-07
	Nuevo	2.95E-11	2.18E-02	7.43E-09
Etapa de excavación - 75 trabajadores	Antiguo	2.15E-08	3.02E-07	3.40E-08
	Nuevo	1.17E-12	9.67E-04	1.07E-09
Etapa de excavación - 80 trabajadores	Antiguo	9.08E-09	2.10E-05	8.65E-09
	Nuevo	1.08E-15	4.57E-02	1.18E-09

Figura 18. Resultados de prueba Shapiro-Wilk sobre las mediciones del simulador antiguo y el simulador nuevo (Elaboración propia).

En la figura se han marcado de color verde las casillas que contengan un valor p menor a 0.05. Es decir, se han marcado de color verde los casos en los que se rechaza la hipótesis nula. De igual forma, se han marcado de color rojo las casillas que tengan un valor mayor a 0.05 y que, por lo tanto, no rechacen la hipótesis nula. Es necesario resaltar que las únicas dos casillas que se han marcado de color rojo se deben a que todos los valores del conjunto de datos evaluados tienen el mismo valor. Por ello, no se pudo calcular su valor p a través de la prueba. Sin embargo, se concluye que, ya que todos los valores son iguales, el conjunto de datos no sigue una distribución normal.

De esta forma, tras analizar todos los conjuntos de datos recopilados en la medición, se

obtiene que todos estos rechazan la hipótesis nula. Es decir, todos estos conjuntos de datos no provienen de una distribución normal. A partir de este hallazgo, se deberá optar por realizar una prueba no paramétrica para obtener conclusiones acerca de la relación entre las características indicadas de los simuladores entre sí.

5.3.2 Definición y comprobación de hipótesis sobre las características de rendimiento

Dado que se trata de conjuntos de datos que no siguen una distribución normal, se optó por evaluarlos a través de una prueba no paramétrica. Asimismo, dado que se buscaba comparar las características de rendimiento indicadas anteriormente para cada configuración del simulador, se optó por utilizar la prueba U-Mann Whitney (Mann & Whitney, 1947). Esta prueba permite determinar si un conjunto de datos es mayor que otro conjunto de datos cuando estos no siguen una distribución normal (Fay & Proschan, 2010, p. 7). De esta forma, se definieron las siguientes hipótesis para realizar la prueba:

H₀: La mediana del valor de la característica es menor en el

simulador antiguo que en el simulador nuevo

H₁: La mediana del valor de la característica es mayor o igual en el

simulador antiguo que en el simulador nuevo

Con estas hipótesis, se realizó la evaluación U-Mann Whitney con un nivel de significancia del 95%. A continuación, se resume el valor p obtenido en cada prueba comparativa por característica de interés.

	Número de hilos	% Procesador	Memoria virtual (Bytes)
Etapa de prospección	2.20E-16	9.96E-01	2.20E-16
Etapa de gestión	2.20E-16	1.00E+00	2.20E-16
Etapa de excavación - 60 trabajadores	2.20E-16	2.20E-16	2.20E-16
Etapa de excavación - 65 trabajadores	2.20E-16	2.20E-16	2.20E-16
Etapa de excavación - 70 trabajadores	2.20E-16	2.20E-16	2.20E-16
Etapa de excavación - 75 trabajadores	2.20E-16	2.20E-16	2.20E-16
Etapa de excavación - 80 trabajadores	8.09E-12	1.23E-11	8.09E-12

Figura 19. Resultados de prueba U-Mann Whitney sobre las mediciones del simulador antiguo y el simulador nuevo (Elaboración propia).

De igual manera que son los resultados de la prueba Shapiro-Wilk, se han marcado de color verde las casillas que tienen un valor p menor a 0.05 y que, por ende, rechazan la hipótesis nula. Por el contrario, se han marcado de color rojo las casillas con un valor p mayor o igual a 0.05 y que, por ello, no pueden rechazar la hipótesis nula. Así, a partir de esta figura, se puede concluir que la mediana de la cantidad de hilos utilizados por el simulador desarrollado en el presente proyecto es menor en todas las configuraciones evaluadas del simulador. Asimismo, se puede concluir, de igual manera, que la mediana de la memoria virtual utilizada por el nuevo simulador desarrollado es menor respecto al simulador original en todas las configuraciones. Sin embargo, no se puede concluir lo mismo acerca del porcentaje de tiempo de uso del procesador al hacer uso del simulador desarrollado. Como se puede apreciar en la figura anterior, el nuevo simulador desarrollado presenta un mayor consumo de CPU durante las fases de prospección arqueológica y gestión de recursos. Esto era de esperarse ya que, tras revisar el código fuente del simulador “El Pez Dorado” original, se encontró una diferencia substancial con respecto al simulador desarrollado. Esta diferencia reside en que, mientras que el simulador original solo realiza el dibujo de los elementos del simulador una sola vez durante cada una de estas dos etapas, el simulador desarrollado realiza un dibujo constante en pantalla. Esta decisión de diseño se tomó en el simulador desarrollado debido a que en la etapa de excavación se necesitaría actualizar constantemente la

imagen mostrada en pantalla debido a, por ejemplo, animaciones o movimiento del cursor. Por esta razón, se optó por realizar constantemente el dibujado en pantalla. Esto explica por qué se tiene un mayor procesamiento en el simulador desarrollado en etapas donde no es necesario volver a realizar un dibujado de la pantalla, como las dos primeras etapas. Por otra parte, esta decisión de diseño también otorga un menor tiempo de uso del procesador en etapas donde sí se requiere un constante dibujado en pantalla, como en la etapa de excavación.

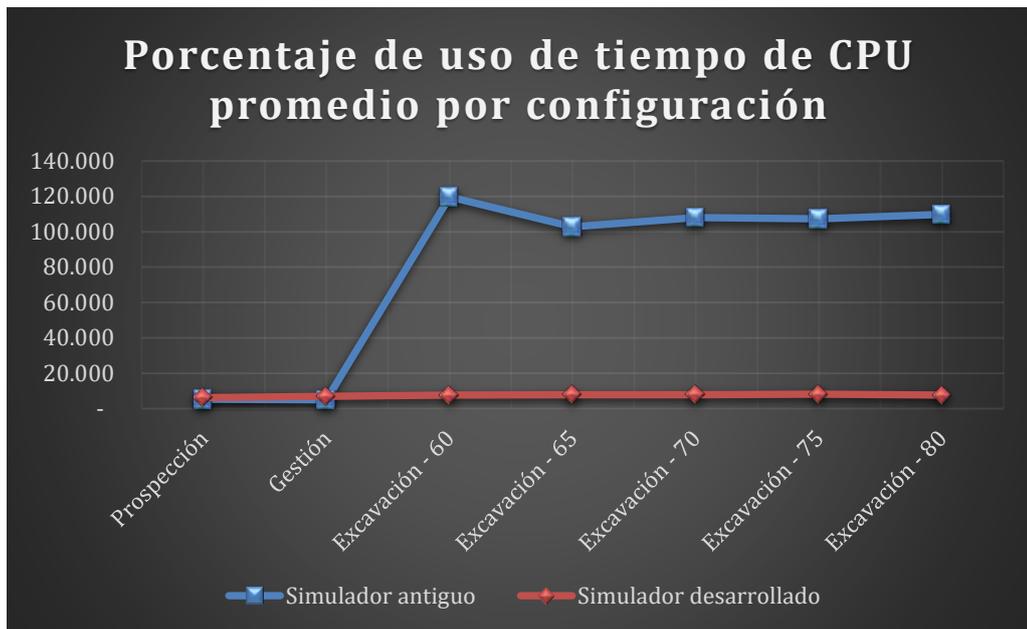


Figura 20. Comparación de porcentaje de uso de tiempo de CPU promedio por configuración (Elaboración propia).

Sin embargo, como se aprecia en la figura 20, el exceso de uso promedio de tiempo de CPU en el simulador desarrollado es mínimo en comparación con el ahorro de este mismo recurso en las configuraciones de la etapa de excavación. La diferencia entre los promedios del simulador desarrollado y del antiguo son, para cada configuración mostrada en la figura y en el mismo orden: -0.90%, -2.06%, 112.05%, 94.93%, 100.19%, 99.10% y 102.25%. Así, se muestra que la pérdida de eficiencia de consumo del tiempo del CPU en las dos primeras configuraciones representa hasta el 0.8% de la ganancia de eficiencia de las siguientes cuatro configuraciones.

De esta manera, se concluye que el simulador desarrollado tiene una mayor eficiencia en el uso de memoria virtual y en el uso de hilos a lo largo de las configuraciones evaluadas. Por otra parte, se concluye también que el simulador desarrollado tiene una

mayor eficiencia en el consumo de tiempo de CPU en la mayoría de las configuraciones, siendo la pérdida en el resto de configuraciones mínima en comparación con la ganancia obtenida.

Respecto a la característica de rendimiento de escalamiento, al igual que en el trabajo de comparación de simuladores utilizado de referencia (Ur Rehman Khan et al., 2013, p. 38), esta se puede evaluar a partir de las figuras 20, 21 y 22.

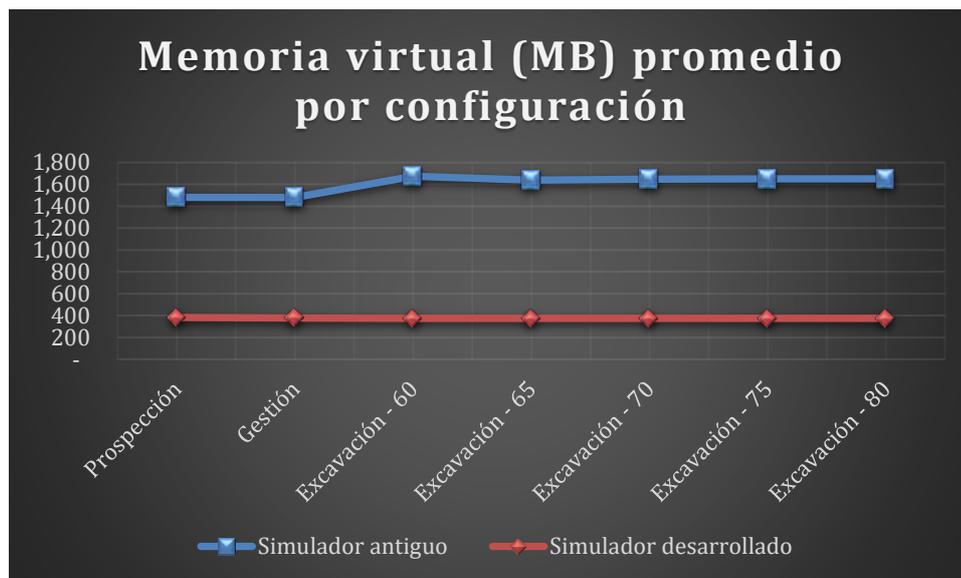


Figura 21. Comparación de consumo de memoria virtual promedio por comparación (Elaboración propia).

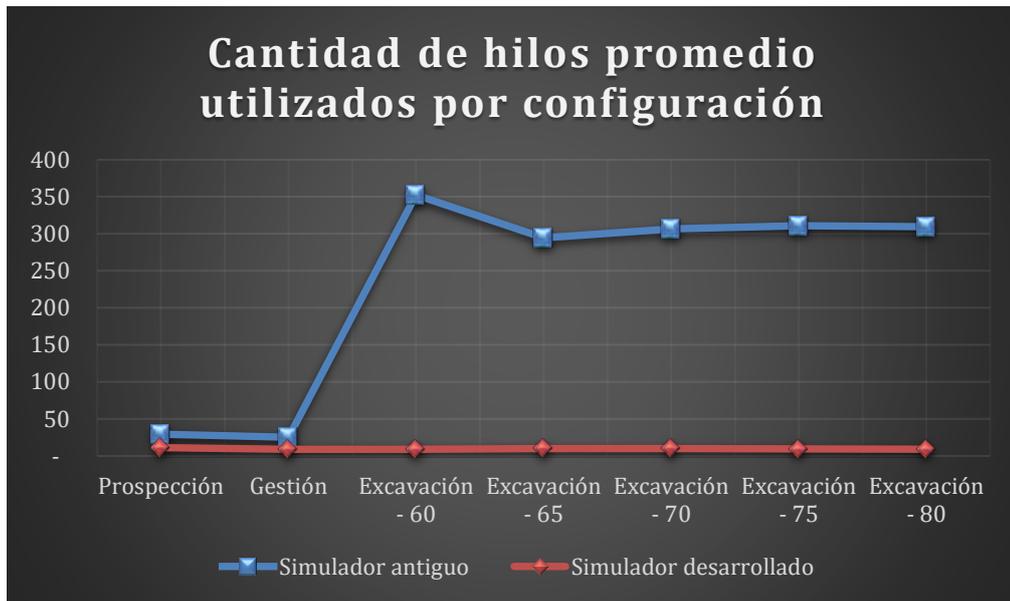


Figura 22. Comparación de cantidad de hilos promedio utilizados por configuración (Elaboración propia).

Como se puede apreciar en las figuras, el comportamiento del consumo de los recursos computacionales indicados no se incrementa rápidamente conforme se incrementa la cantidad de trabajadores en el simulador. Dado que se ha considerado un incremento uniforme de trabajadores desde la cantidad mínima y máxima de estos y a que no se ha incrementado significativamente el consumo de recursos, se concluye que el simulador desarrollado cumple la característica de escalabilidad.

Para resumir, en esta sección se ha concluido a través de experimentación numérica que el simulador desarrollado cumple tiene un consumo más eficiente en comparación del simulador original de los recursos computacionales memoria virtual, tiempo de uso del CPU y número de hilos utilizados. Asimismo, a partir de las gráficas de consumo de estos recursos, adicionalmente se ha concluido que el simulador desarrollado cumple la característica de escalabilidad.

Capítulo 6. Desarrollo de un simulador de excavaciones arqueológicas en base a los aspectos identificados

En este capítulo se detallarán las acciones realizadas para cumplir con el resultado esperado 6 “Desarrollar un simulador de excavaciones arqueológicas en base a los aspectos identificados”. Como se describió en la sección 1.2.4. “Mapeo de objetivos, resultados y verificación”, este objetivo específico se considerará alcanzado a través del cumplimiento de los siguientes dos resultados esperados:

R5. Módulo de simulación del proceso de excavación arqueológica

R6. Módulo de interfaz de usuario del simulador de excavación arqueológica

De esta manera, el presente capítulo se dividirá en dos secciones: una sección para cada resultado esperado descrito. Es necesario resaltar que, de acuerdo a la metodología de desarrollo a seguir, cada uno de estos módulos se compondrá de tres iteraciones de desarrollo. Al final de cada una de ellas se realizarán una prueba de usabilidad con usuarios que pertenecen a uno de los dos grupos de Personas identificadas en el objetivo específico 1. Asimismo, se recibirán observaciones del simulador en estas iteraciones para ser subsanadas en la siguiente prueba de usabilidad.

6.1 Desarrollo del módulo de simulación del proceso de excavación arqueológica

En esta sección, se describirán las acciones llevadas a cabo para el cumplimiento del resultado esperado 5, “Módulo de simulación del proceso de excavación arqueológica”. Como se explicó en la sección anterior, el desarrollo de cada módulo del simulador se dividió en tres iteraciones. Esta cantidad se determinó a partir del trabajo de Merino, Aguilar y Zapata (Merino, Zapata, & Aguilar, 2017) y a partir de la estimación de las complejidad de las tareas requeridas.

De esta manera, se dividió el trabajo a realizar en las tres iteraciones en las siguientes actividades:

1. Primera iteración

- a. Implementar el bucle principal del simulador.
- b. Implementar el cargado de recursos externos.
- c. Implementar las escenas del simulador y la interacción entre ellas.
- d. Implementar la finalización de la simulación, durante la etapa de excavación.

- e. Implementar el menú de juego.
 - f. Implementar el método de muestreo aleatorio.
 - g. Implementar la finalización y resumen de la etapa de prospección.
 - h. Implementar la asignación de cantidades por defectos de recursos a cada yacimiento, durante la etapa de gestión de recursos.
 - i. Implementar las restricciones en la etapa de prospección para pasar a la etapa de gestión de recursos.
2. Segunda iteración
- a. Implementar el ingreso de recursos a un yacimiento durante la etapa de gestión de recursos.
 - b. Implementar el guardado y carga de recursos asignados a un yacimiento durante la etapa de gestión de recursos.
3. Tercera iteración
- a. Implementar el método de muestreo estratificado.
 - b. Implementar el método de muestreo no aleatorio.
 - c. Implementar el comienzo de excavación de uno o todos los yacimientos.
 - d. Implementar la detención de las excavaciones en uno o todos los yacimientos.
 - e. Implementar el comportamiento de los trabajadores durante la etapa de excavación.
 - f. Implementar el registro de artefactos recuperados en el inventario.
 - g. Implementar la adquisición de nuevos recursos durante la etapa de excavación.
 - h. Implementar el paso del tiempo durante la etapa de excavación.

A partir de la identificación de las tareas a realizarse, se procedió a desarrollar la simulación del proceso en base a las actividades de cada iteración.

Respecto al presente módulo, el desarrollo se enfocó en la inicialización y modificación o actualización del estado interno de cada escena, de cada estado que componen las escenas y de cada variable miembro que componen los estados. Así, según la arquitectura presentada en el objetivo específico 2, el desarrollo se enfocó en la implementación de las funciones “*onEnter*” y “*update*” que componen cada clase “*Scene*” y cada clase “*GameState*”. Asimismo, el desarrollo de este módulo incluyó el desarrollo de cualquier clase adicional que fuera necesario para implementar el

comportamiento de las clases “*GameState*”, como las clases “*Button*”, “*Map*”, o “*HumanResource*”.

Es necesario resaltar que las actividades identificadas en cada iteración no se desarrollaron por separado de las actividades del módulo de interfaz de usuario. Por el contrario, debido a su acoplamiento por utilizar los mismos elementos del simulador, se desarrollaron en simultáneo.

Finalmente, al terminar las actividades del módulo de simulación del proceso de excavación arqueológica, se obtuvo un producto de software que actualizaba con éxito el estado interno de sus elementos.

6.2 Desarrollo del módulo de interfaz de usuario del simulador de excavaciones arqueológicas

En esta sección, se describirán las acciones llevadas a cabo para el cumplimiento del resultado esperado 6, “Módulo de interfaz de usuario del simulador de excavaciones arqueológicas”.

Al igual que en el desarrollo del módulo de simulación del proceso de excavación, se definió una lista de actividades que se iban a desarrollar a lo largo de las tres iteraciones de desarrollo del proyecto. Es necesario indicar que en este módulo se consideró el desarrollo de la parte gráfica de todas las actividades listadas en el módulo de simulación. A continuación, se presentan solo las nuevas actividades no consideradas en la lista de actividades presentadas en el módulo anterior.

1. Primera iteración

- a. Implementar la selección de un yacimiento durante la etapa de excavación.
- b. Implementar la visualización de la información de un yacimiento durante la etapa de excavación.

2. Segunda iteración

- a. Implementar la visualización del detalle de cada tipo de recurso durante la etapa de gestión de recursos.
- b. Implementar el marcado de cada yacimiento al que ya se le han asignado recursos, durante la etapa de gestión de recursos.
- c. Implementar el incremento del tamaño o “*zoom*” del mapa, durante la etapa de excavación.

3. Tercera iteración

- a. Implementar la notificación de artefacto encontrado, durante la etapa de excavación.
- b. Implementar el mini mapa y sus interacciones, durante la etapa de excavación.
- c. Implementar la notificación de finalización de excavación en un nivel.
- d. Implementar las animaciones de los obreros durante la etapa de excavación.
- e. Implementar la variación de los gráficos de un yacimiento según el nivel en que se encuentre.
- f. Implementar la notificación y resumen de la semana al final de esta, durante la etapa de excavación.
- g. Implementar el uso de un gráfico en lugar del cursor por defecto.

Una vez definidas la lista de actividades a llevarse a cabo, se procedió al desarrollo de cada actividad del módulo. Este desarrollo se enfocó, a diferencia del desarrollo del módulo de simulación del proceso de excavación, en las funciones “*render*” de las clases “*Scene*” y “*State*”. Asimismo, se hizo extenso uso de las clases “*TextureManager*” y “*FontManager*” para acceder a las texturas y fuentes, respectivamente, que se iban a dibujar en pantalla.

Respecto al origen de estos dos recursos, estos se obtuvieron directamente del simulador original. Sin embargo, como se comentará posteriormente en el objetivo específico 4, fue necesario realizar modificaciones a la interfaz gráfica tras realizar las pruebas de usabilidad. Por esta razón, el autor del presente proyecto modificó algunas imágenes para crear nuevas texturas que se ajusten a los requerimientos de usabilidad. Ejemplo de esto es que se agregaron nuevas texturas de fuentes hídricas en el mapa original.

Finalmente, al finalizar el desarrollo, se obtuvo un producto de software que mostraba la interfaz de usuario que permitía interactuar con el simulador y, así, permitía realizar la simulación.

Capítulo 7. Validar la usabilidad del simulador desarrollado con una muestra de estudiantes

En el presente capítulo, se describirán las acciones que se tomaron para cumplir con el objetivo específico 7 del presente proyecto.

Este objetivo específico, a diferencia de los anteriores, solo cuenta con un único resultado esperado. Este resultado se detalla en la siguiente sección.

7.1 Resultado de pruebas de usabilidad de la interfaz de usuario del simulador desarrollado.

Las pruebas de usabilidad siguieron la metodología utilizado en el trabajo de Daly, Zapata y Paz (Daly, Zapata, & Paz, 2011). De esta manera, se planificaron cuatro pruebas de usabilidad a lo largo del proyecto: Una al finalizar cada iteración de desarrollo y una prueba adicional al finalizar el desarrollo. Las pruebas al final de cada iteración evaluarían la usabilidad de las tareas implementadas durante la iteración. Por otro lado, la prueba adicional al finalizar el desarrollo serviría para realizar una evaluación general de la usabilidad de todo el simulador desarrollado.

De esta forma, las pruebas de usabilidad tuvieron como objetivos:

1. Primera iteración
 - a. Identificar el nivel de claridad de los mensajes mostrados al usuario durante la etapa de prospección arqueológica
 - b. Identificar el nivel de comprensión de los pasos necesarios
 - c. Identificar el nivel de claridad de los mensajes mostrados al usuario durante la etapa de excavación arqueológica
 - d. Identificar el nivel en el cual el lenguaje utilizados en los mensajes mostrados al usuario se sentían naturales.
2. Segunda iteración
 - a. Identificar el nivel de claridad de los mensajes mostrados al usuario durante la etapa de gestión de recursos
 - b. Identificar el nivel de comprensión de las descripciones de los recursos mostrados durante la etapa de gestión de recursos.
 - c. Identificar el nivel en el cual la interacción con los botones de la etapa de gestión de recursos se sintió naturales.

3. Tercera iteración

- a. Identificar el nivel de claridad de los mensajes mostrados durante la etapa de prospección.
- b. Identificar el nivel de comprensión de la descripción de los métodos de muestreo mostrados durante la etapa de prospección.
- c. Identificar el nivel de claridad de los mensajes mostrados durante la etapa de excavación arqueológica.
- d. Identificar el nivel de comprensión de las acciones que realizan los botones “de acción” durante la etapa de excavación arqueológica.
- e. Identificar el nivel de comprensión de las restricciones para empezar a excavar un yacimiento durante la etapa de excavación arqueológica.

4. Prueba de usabilidad final

- a. Todos los objetivos anteriores.

De esta manera, para alcanzar estos objetivos y según la metodología utilizada en el trabajo indicado anteriormente, se realizaron pruebas de usabilidad con la siguiente estructura:

1. El usuario escogido completaba un cuestionario sobre sus conocimientos previos sobre el proceso de excavación arqueológica.
2. El usuario utilizaba el simulador mientras realizaba la técnica de usabilidad de pensar en voz alta (Rubin & Chisnel, 2014, p. 92). En simultáneo, el evaluador tomaba apuntes sobre una lista de tareas que el usuario debía realizar.
3. El usuario respondía un cuestionario sobre su experiencia utilizando el simulador.

La estructura de cada evaluación en específico puede ser encontrada en el anexo I. Respecto a los usuarios sobre los que se aplicó esta evaluación, se escogieron a tres alumnos de educación superior, según el trabajo anteriormente citado. De estos tres alumnos, uno correspondía al grupo de personas 1, detallado en el objetivo específico 1, y dos correspondían al grupo de personas 2. De esta manera, se esperaba realizar las evaluaciones con personas que tengan el mismo perfil que los usuarios objetivo del simulador.

Finalmente, tras realizar las evaluaciones de usabilidad al final de cada iteración, se utilizaron las respuestas de los cuestionarios y los apuntes del evaluador para generar

una nueva lista de tareas a desarrollar para incrementar la usabilidad del simulador. Esta lista se menciona a continuación:

1. Etapa de prospección arqueológica

- a. Explicar claramente el concepto de prospección
- b. Explicar la existencia del menú del juego y cómo acceder a él.
- c. Indicar que no se pueden seleccionar celdas con agua cuando se selecciona el método de muestreo no aleatorio.
- d. No permitir que se vuelva a ejecutar un método de muestreo que ya se ha seleccionado, ya que confunde al usuario.
- e. Permitir que se pueda escoger un método de muestreo no solo presionando la caja para marcar al costado del nombre, sino también presionando el nombre.

2. Etapa de gestión de recursos

- a. Explicar que se han desmarcado los sitios donde no se han encontrado los yacimientos.
- b. Explicar la diferencia entre bandera blanca y roja.
- c. Explicar, en el mensaje de introducción, que hay cantidades mínima y máxima de los recursos.
- d. Explicar que el proyecto va a tener una duración máxima de cuatro semanas.
- e. Mostrar el presupuesto del proyecto desde esta etapa.
- f. Cambiar el nombre de "Rasqueta" a "Badilejo".
- g. Cambiar la descripción negativa del recurso "Estudiante".
- h. Cambiar la moneda del juego de soles a créditos.

3. Etapa de excavación arqueológica

- a. No mencionar la destrucción de artefactos. En su lugar, mencionar afectación.
- b. Agregar texto cuando se pasa encima de los botones de acción para explicar qué acción realizan.
- c. Explicar las condiciones de inicio de excavación.
- d. Explicar la necesidad de utilizar un balde junto con las otras herramientas.
- e. Arreglar la posición del cursor cuando se arrastre un objeto, de manera que el objeto quede en la punta del cursor.

- f. Explicar la posibilidad de afectación de cada herramienta y la magnitud de esta posibilidad por cada herramienta.
- g. Explicar la velocidad de excavación de cada herramienta y la magnitud de esta velocidad por cada herramienta.
- h. Resaltar los mensajes de recuperación o afectación de los artefactos encontrados.
- i. Facilitar la interacción con la barra deslizadora.
- j. Explicar que se va a detener la excavación de un yacimiento en cuanto se finalice un nivel.
- k. Explicar qué es el inventario y que se trata de artefactos recuperados y no de herramientas.

Tras corregir las observaciones de usabilidad descritas, se realizó la prueba de usabilidad final de todo el simulador. Los resultados de esta prueba comprobaron el incremento de la usabilidad del simulador.

De esta manera, se realizaron las pruebas de usabilidad al final de cada iteración y se tomaron en consideración las observaciones levantadas para incrementar la usabilidad del simulador. El incremento de la usabilidad del simulador fue comprobada a través de la evaluación final de usabilidad de todo el simulador.

Capítulo 8. Conclusiones y trabajos futuros

En el presente trabajo, se desarrolló un simulador de excavaciones arqueológicas acorde a los principios académicos que se enseñan en los centros educativos de educación superior. Asimismo, se consideró, durante el desarrollo del simulador, el rendimiento a partir de su arquitectura de software y la usabilidad que este debía asegurar. En este capítulo, se presentan las conclusiones que se han tenido tras el desarrollo del simulador y las recomendaciones de trabajos futuros.

8.1 Conclusiones

En esta sección, se presentarán las conclusiones a las que se han llegado tras el cumplimiento de cada resultado esperado.

- i. Documento de análisis de requisitos con los aspectos del proceso de excavación arqueológica que deben ser incluidos en el simulador de excavaciones arqueológicas

En este resultado esperado se identificaron y definieron los requisitos que debía cumplir el simulador a desarrollarse. Con esta meta, se entrevistó a una experta en la docencia de arqueología con experiencia en campo sobre las excavaciones arqueológicas. A pesar de que se hizo una investigación previa sobre el proceso de excavación, realizar una entrevista con un experto en arqueología fue de gran importancia ya que permitió identificar las diferencias entre la práctica teórica y real de las excavaciones arqueológicas.

Asimismo, como parte de la metodología seguida y descrita, se identificaron personas y escenarios que debían considerar los requisitos. Esta herramienta para definir los requisitos fue de gran utilidad ya que permitió considerar posibles problemas de usabilidad que tendrían los futuros usuarios del sistema desde la fase de análisis.

- ii. Validación académica de los aspectos a ser incluidos en el simulador de excavaciones arqueológicas a través de evaluación por juicio experto de un profesional en la docencia de arqueología

Para cumplir este resultado esperado, se formularon prototipos en papel y digitales en base a los requisitos identificados en el resultado esperado anterior. Posteriormente, se presentaron estos prototipos al experto para validar el contenido académico. Tras haber

realizado esta prueba, se concluye que realizar prototipos digitales es más útil que realizar prototipos en papel. Esto se debió a que el prototipo digital exige mayor nivel de detalle en las imágenes que el prototipo en papel. Esto tuvo como resultado que el experto consultado pudo brindar observaciones sobre detalles en el prototipo digital que no se encontraban en los prototipos en papel. Por otra parte, los prototipos en digital también resultan más útiles debido a que se pueden reutilizar en la etapa de desarrollo. Esto se debe a que las características, como tamaño, posición y color, de los recursos utilizados en el prototipo, como imágenes y fuentes, se mantuvieron en el producto final. De esta manera, se pudo reutilizar las configuraciones realizadas en el prototipo en el producto final y, así, se pudo incrementar la velocidad del desarrollo.

iii. Documento de arquitectura de software que describa la arquitectura que soporte la simulación de excavaciones arqueológicas

Para el cumplimiento de este resultado esperado, se diseñó una arquitectura de software enfocada al consumo eficaz de los recursos computacionales y se le representó a través del modelo de vista de arquitectura 4+1. Para el diseño, se realizó una búsqueda previa sobre patrones de diseño utilizados en software similares, específicamente en video juegos. Estos patrones resultaron de gran utilidad durante el diseño, ya que permitieron considerar problemas de arquitectura que aún no habían surgido. De esta manera, se concluye que la búsqueda de patrones de diseño, aún en industrias similares, pueden ser una inversión de tiempo beneficiosas, ya que se evita invertir tiempo en resolver problemas de diseño comunes.

iv. Resultado de pruebas de estrés del simulador desarrollado con la arquitectura diseñada y que demuestre un mejor rendimiento que el simulador “El Pez Dorado”

Como parte de este resultado esperado, se realizaron mediciones acerca de recursos computacionales durante la ejecución tanto del simulador desarrollado como el simulador original, según un trabajo comparativo de simuladores similar (Ur Rehman Khan et al., 2013). A partir de estas, se realizó una experimentación numérica para determinar qué simulador tiene un menor consumo de los recursos computacionales en cada una de las configuraciones evaluadas.

Tras haber concluido este resultado esperado, se concluye que el uso de una metodología de comparación de trabajos similares puede ser más informativa que desarrollar una propia metodología. Esto se debe a que, en primera instancia, no se

había considerado la evaluación de ambos simuladores en distintas configuraciones. Solo se había considerado la comparación de las mediciones obtenidas al seguir el flujo completo del simulador y no por etapas. Sin embargo, el hacer uso de mediciones por configuraciones permitió descubrir comportamientos que no se hubiesen detectados si no se hubiesen utilizado configuraciones. Este es el caso del mayor consumo de tiempo del CPU en las etapas de prospección y gestión. Esto ocasionó que se analizaran comparativamente los códigos fuente del simulador desarrollado y del simulador original. Esto llevó a determinar las razones por las que se debía este comportamiento y, así, tener una mejor comprensión de los resultados de las comparaciones.

Asimismo, dado que se comprobó que el nuevo simulador desarrollado, el cual presenta la misma simulación aunque con una arquitectura distinta, tiene un menor consumo de recursos computacionales (tiempo de utilización de CPU, memoria RAM y número de hilos), se concluye que se desarrolló un simulador más eficiente debido a su arquitectura.

v. Módulo de simulación del proceso de excavación arqueológica

Como parte del cumplimiento de este resultado esperado, se dividió el trabajo a desarrollar en múltiples actividades. Posteriormente, estas actividades fueron divididas en tres iteraciones de desarrollo. A pesar de que el orden en el cual se desarrollaron las actividades fue correcto en términos de dependencias entre actividades, no fue el adecuado desde la perspectiva de usuario. Esto se debe a que la segunda iteración se enfocó exclusivamente en la etapa de gestión de recursos y la tercera iteración se enfocó principalmente en la etapa de excavaciones. De esta manera, no fue hasta la finalización de todo el desarrollo cuando se obtuvo un flujo completo utilizable por el usuario. Esto influyó, posteriormente, en las pruebas de usabilidad que se realizaron al final de cada iteración, dado que el usuario no lograba comprender la finalidad de las etapas que se habían desarrollado mínimamente. Por estos motivos, se concluye que se debe tener en igual consideración el criterio de la perspectiva del usuario como la dependencia entre actividades al realizar la división del desarrollo en iteraciones.

vi. Módulo de interfaz de usuario del simulador de excavaciones arqueológicas

Durante este resultado esperado, se desarrollaron los componentes de software involucrados con la interfaz de usuario. Respecto a los recursos utilizados en la interfaz, estos se obtuvieron del simulador original. Sin embargo, se concluye que se debe tener

en cuenta un riesgo adicional al reutilizar recursos de otros proyectos como imágenes: la desaprobación de los recursos a utilizarse por parte de los usuarios en términos de usabilidad. Esto se debió a que, como se detectaron en la prueba de usabilidad, el mapa principal del simulador no permitía a los usuarios comprender el método de muestreo estratificado. Por ello, se crearon nuevos recursos para ser usados como mapa que sí cumplieran con este criterio. Por esta razón, se concluye que se debe tener en cuenta este tipo de riesgo durante la fase de planeación del proyecto.

vii. Resultados de pruebas de usabilidad de la interfaz de usuario del simulador desarrollado

A pesar de que a lo largo de la carrera el presente alumno ya había recibido información sobre la importancia de la usabilidad, llevar a cabo este resultado esperado fue revelador sobre la necesidad de hacer pruebas de usabilidad. Esto se debe a que, como se mostró a lo largo de las doce pruebas realizadas, el usuario final no lograba tener una comprensión cabal sobre cómo realizar una simulación eficazmente. En cada prueba se logró determinar los problemas que encontraba el usuario para continuar con el flujo de la simulación.

Asimismo, otro hecho de importancia que se determinó durante la realización de este resultado esperado es que el usuario puede indicar explícitamente un conjunto de problemas que ha encontrado, pero puede omitir otros problemas por no saber cómo formularlos. Por ello, se concluye que es de igual importancia observar al usuario mientras hace uso del software y registrar los problemas que pueda presentar.

8.2 Trabajos futuros

En esta sección, se detallarán las propuestas de trabajos futuros identificadas tras realizar el presente proyecto.

i. Editor de mapas

A pesar de que el mapa utilizado a lo largo de la simulación contiene las características necesarias para comprender los métodos de muestreo, se podría desarrollar una herramienta que permita modificarlo. De esta manera, se podría apreciar la variación de

los métodos de muestreos a partir de los diferentes terrenos y climas que se encuentran en la práctica de arqueología.

ii. Editor de artefactos

Al igual que el mapa utilizado, la cantidad y variedad de artefactos encontrados en el terreno excavado es constante. Sin embargo, se recomienda el desarrollo de una herramienta que permita introducir nuevos tipos de artefactos. Esto serviría para concientizar a los usuarios sobre la variedad de artefactos que se podrían encontrar en campo y, así, introducir información sobre las herramientas más adecuadas para trabajar con ellos.

iii. Incorporar imágenes en las pantallas de explicación

Las pantallas de explicación mostradas al inicio de cada etapa del proyecto arqueológico solo cuentan con texto. En un trabajo posterior, se podrían añadir imágenes de los botones o personajes con los que el usuario puede interactuar. De esta manera, el usuario comprendería mejor los pasos que debe tomar en cada etapa.

iv. Selección de género del usuario

De acuerdo con la experta de la docencia de arqueología, la mayoría de los egresados de aquella carrera son mujeres. Por ello, permitirle al usuario seleccionar su género al inicio de la simulación y mostrar personajes con el género seleccionado haría que la simulación sea más realista.

Capítulo 9. Referencias

Abdo, A. M., & Alali, S. H. (2016). Comparing Common Programming Languages to Parse Big XML File in Terms of Executing Time, Memory Usage, CPU Consumption and Line Number on Two Platforms. *European Scientific Journal*, 12(27). Consulta: 12 de Mayo de 2022.

<https://doi.org/10.19044/esj.2016.v12n27p325>

Aguilar, M., & Zapata, C. (2017). Integrating UCD and an Agile Methodology in the Development of a Mobile Catalog of Plants BT - Advances in Ergonomics Modeling, Usability & Special Populations. *Advances in Ergonomics Modeling, Usability & Special Populations: Proceedings of the AHFE 2016 International Conference on Ergonomics Modeling, Usability & Special Populations, July 27-31, 2016, Walt Disney World®, Florida, USA, 75–87.*

Akpata, E., & Riha, K. (2004). Can Extreme Programming be used by a Lone Programmer? *Systems Integration*, 167–175. Consulta: 12 de Mayo de 2022.

<http://munit.net/AkpataRihaEXSI2004.pdf>

Archaeological Institute of America. (2007). *Archaeology* 101, 1–4. Consulta: 12 de Mayo de 2022.

<https://www.archaeological.org/pdfs/education/Arch101.2.pdf>

Archaeology Education Clearinghouse. (n.d.). National Council for the Social Studies. Consulta: 30 de abril de 2022

<https://archaeologyeducationclearinghouse.wordpress.com/resources/intro/>

Arnaldo, R., Perez, L., Crespo, J., & Alonso, F. (2011). The use of virtual flight simulation for airspace design in aeronautical engineering education. *2011 IEEE Global Engineering Education Conference, EDUCON 2011*, 543–551. Consulta:

12 de Mayo de 2022.

<https://doi.org/10.1109/EDUCON.2011.5773191>

Ashmore, P., & Baumann, T. (2003). Teacher'S Corner: Archaeology for the Classroom. Consulta: 30 de Abril de 2022

<https://silo.tips/download/teacher-s-corner-archaeology-for-the-classroom-by-pamela-ashmore-and-timothy-bau>

Bacca, J. L., Molina, J. A., Cavanzo, G. A., & Guevara, J. C. (2009). Educational simulator of conic sections supported by intelligent agents and semantic web: An example in the development and design of educational simulators for mathematics. *Proceedings - Frontiers in Education Conference, FIE*, 1–6.

Consulta: 2 de Mayo de 2022.

<https://doi.org/10.1109/FIE.2009.5350820>

Baldeón, J. (2010). DESARROLLO DE UN VIDEOJUEGO APLICADO A LA ENSEÑANZA DE ARQUEOLOGÍA.

Banks, J., Nelson, B. L., Carson, J. S., & Nicol, D. M. (2004). *Discrete-Event System Simulation. PrenticeHall international series in industrial and systems engineering* (Cuarta edición). Prentice Hall. Consulta: 2 de Mayo de 2022.

<https://doi.org/10.2307/1268124>

Barker, G., Roberts, C., Gosden, C., Horning, A., & Welham, K. (2016). *Reflections on Archaeology*. Consulta: 2 de Mayo de 2022.

<https://www.britac.ac.uk/reflections-on-archaeology>

Beck, K., & Andres, C. (2004). *Extreme Programming Explained: Embrace Change. Extreme Programming Explained* (2nd editio). Addison-Wesley. Consulta: 2 de Mayo de 2022.

<http://ptgmedia.pearsoncmg.com/images/9780321278654/samplepages/9780321278654.pdf>

Brown, S. (2011). Trash Talks. *ARCHAEOLOGICAL INSTITUTE of AMERICA*, 1–2.
Consulta: 2 de Mayo de 2022.

[https://www.archaeological.org/sites/default/files/files/Trash Talks-edited.pdf](https://www.archaeological.org/sites/default/files/files/Trash%20Talks-edited.pdf)

Capuz, M. A., Policarpio, L., Quinto, B. J., Reyes, S. E., & Uy, R. L. (2016). MIPS64 simulator on a mobile environment. *Proceedings - 2016 IEEE Region 10 Symposium, TENSYP 2016*, 219–224. Consulta: 2 de Mayo de 2022.

<https://doi.org/10.1109/TENCONSpring.2016.7519408>

Croucher, K., Cobb, H., & Brennan, A. (2008). *Investigating the role of fieldwork in teaching and learning archaeology*. Consulta: 23 de Mayo de 2022.

<https://www.advance-he.ac.uk/knowledge-hub/investigating-role-fieldwork-teaching-and-learning-archaeology>

Daly, C., Zapata, C., & Paz, F. (2011). Improving the usability in a video game through continuous usability evaluations.

Fay, M. P., & Proschan, M. a. (2010). Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules.

Stat Surv, 4, 1–39. Wilcoxon-Mann-Whitney. Consulta: 2 de Mayo de 2022.

<https://doi.org/10.1214/09-SS051>.

Garcia, J. (2011). A Web Based Simulator on Elementary Queuing.

En *IEEE Xplore*. Consulta: 2 de Mayo de 2022.

<https://ieeexplore.ieee.org/abstract/document/1358142>

Getchell, K., Miller, A., Nicoll, J. R., Sweetman, R. J., & Allison, C. (2010). Games methodologies and immersive environments for virtual fieldwork. *IEEE*

Transactions on Learning Technologies, 3(4), 281–293. Consulta: 6 de Mayo de 2022.

<https://doi.org/10.1109/TLT.2010.25>

Ghazali, K. W. M., Hassan, R., & Ali, Z. (2013). A Network Device Simulator in Android Platform. *Advanced Communication Technology (ICACT), 2013 15th International Conference On*, 378–381. Consulta: 6 de Mayo de 2022.

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6488210&isnumber=6488107>

Huitt, W. (2003). Classroom Instruction. *Educational Psychology Interactive*, (Valdosta State University), 18202. Consulta: 6 de Mayo de 2022.

<https://doi.org/10.1002/trtr.1286>

Imai, Y., & Takeichi, K. (2015). Development and Evaluation of Adobe Flash Based CPU Scheduling Simulator Executable on Major Multiple Web Browsers.

Proceedings - 2015 International Conference on Intelligent Networking and Collaborative Systems, IEEE INCoS 2015, 149–155. Consulta: 6 de Mayo de 2022.

<https://doi.org/10.1109/INCoS.2015.92>

Ines, D., Burke, H., & Smith, C. (2007). *Manual de campo del arqueólogo*. Barcelona: Ariel.

International Organization for Standardization. (1998). Iso 9241-11. *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) -- Part 11: Guidance on Usability*, 1998, 22.

Jánó, R., Fodor, A., & Ilieş, A. I. (2016). Simulation and Educational Tool for Fault Modelling in Logic Circuits, (Vcc), 476–479. Consulta: 6 de Mayo de 2022.

<https://ieeexplore.ieee.org/document/7563244>

Johnson, M. (2010). *Archaeology Theory. An introduction* (2nd editio). Blackwell Publishing.

Kauppa. (2009). Teaching Activity: Every Picture Tells a Story. Consulta: 6 de Mayo de 2022.

<https://web.archive.org/web/20181128225133/https://anthropology.si.edu/outreach/pdfs/Every%20Picture%20Tells%20a%20Story.pdf>

Kazman, R., Abowd, G., Bass, L., & Clements, P. (1996). Scenario-based analysis of software architecture. *IEEE Software*, 13(6), 47–55. Consulta: 6 de Mayo de 2022. <https://doi.org/10.1109/52.542294>

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(TR/SE-0401), 28. Consulta: 6 de Mayo de 2022.

https://www.researchgate.net/publication/228756057_Procedures_for_Performing_Systematic_Reviews

Kruntchen, P. (1995). Architectural blueprints—the” 4+ 1” view model of software architecture. *IEEE Software*, 12(November), 42–50. Consulta: 6 de Mayo de 2022. <https://doi.org/10.1145/216591.216611>

Lambertsen, R. T., Tang, S., Davies, J., & Morecroft, C. (2017). Serious Gaming for Pharmacy Education: Development of a Serious Games for Teaching Pharmacist Communication and Drug Administration in a Virtual Hospital Setting. *Proceedings - 2016 9th International Conference on Developments in ESystems Engineering, DeSE 2016*, 151–156. Consulta: 6 de Mayo de 2022.

<https://doi.org/10.1109/DeSE.2016.25>

Lee, W. N., Jeong, Y. W., Park, J. B., Shin, J. R., & Lee, K. Y. (2007). Development of

an educational simulator for particle swarm optimization and economic dispatch applications. *2007 International Conference on Intelligent Systems Applications to Power Systems, ISAP*. Consulta: 6 de Mayo de 2022.

<https://doi.org/10.1109/ISAP.2007.4441693>

Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. Consulta: 6 de Mayo de 2022.

[https://medstatistic.ru/articles/Mann and Whitney 1947.pdf](https://medstatistic.ru/articles/Mann%20and%20Whitney%201947.pdf)

Mattsson, M., Grahn, H., Mårtensson, F., Box, P. O., & Ronneby, S.-. (2006). Software Architecture Evaluation Methods for Performance , Maintainability , Testability , and Portability. Consulta: 6 de Mayo de 2022.

https://www.researchgate.net/publication/200622064_Software_Architecture_Evaluation_Methods_for_Performance_Maintainability_Testability_and_Portability

McLoone, H. E., Jacobson, M., Hegg, C., & Johnson, P. W. (2010). User-centered design. *Work*, 37(4), 445–456. Consulta: 6 de Mayo de 2022.

https://www.researchgate.net/publication/49630133_User-centered_design_and_evaluation_of_a_next_generation_fixed-split_ergonomic_keyboard

Merino, E., Zapata, C., & Aguilar, M. del C. (2017). UCD and Agile Methodology in the Development of a Cultural Heritage Platform, 1–26.

Moreira, a, Haller, J., & Hansson, H. (2013). *Sfml Game Development*. Consulta: 6 de Mayo de 2022.

<http://books.google.com/books?hl=en&lr=&id=4oYIKBI5g24C&oi=fnd&pg=PT10&dq=SFML+Game+Development&ots=1K1Don1Sht&sig=4vxufAhdGE2nxSnrp2AHn-fk9Fw>

- Noll, J., Butterfield, A., Farrell, K., Mason, T., McGuire, M., & McKinley, R. (2014). GSD Sim: A Global Software Development Game. *Proceedings - International Computer Software and Applications Conference, 18–21–Augu, 15–20*. Consulta: 6 de Mayo de 2022.
<https://doi.org/10.1109/ICGSEW.2014.12>
- Nystrom, R. (2014). *Game Programming Patterns*. Genever Benning.
- Oladosu, J. B., Adigun, M. O., & Emuoyibofarhe, J. O. (2013). A mobile virtual patient for medical learning. *2013 Pan African International Conference on Information Science, Computing and Telecommunications (PACT), 74–79*. Consulta: 6 de Mayo de 2022.
<https://doi.org/10.1109/SCAT.2013.7055094>
- Oztekin, H., Temurtas, F., & Gulbag, A. (2010). BZK.SAU: Implementing a hardware and software-based computer architecture simulator for educational purpose. *2010 International Conference on Computer Design and Applications, ICCDA 2010, 4(Iccda), 90–97*. Consulta: 6 de Mayo d 2022.
<https://doi.org/10.1109/ICCDA.2010.5541476>
- Paramita, A., & Smitha, K. G. (2017). PARACACHE : Educational Simulator for Cache and Virtual Memory, 234–238. Consulta: 6 de Mayo de 2022.
<https://doi.org/10.1109/ISET.2017.60>
- Prechelt, L. (2000). An empirical comparison of seven programming languages. *Computer, 33(10), 23–29*. Consulta: 6 de Mayo de 2022.
<https://doi.org/10.1109/2.876288>
- Quality Assurance Agency for Higher Education. (2014). Subject Benchmark Statement: Archaeology, (December), 2014. Consulta: 6 de Mayo de 2022

<https://dera.ioe.ac.uk/21794/1/SBS-archaeology-2014.pdf>

Rauen, Z. I., Rolfe, N. S., Kantarci, B., Banavar, M., & Freitag, W. (2016). A Tool for Simulation and Visualization of Distributed Estimation in Wireless Sensor Networks, 34–36. Consulta: 10 de Mayo de 2022.

<https://ieeexplore.ieee.org/document/7457550>

Reilly, P. (1991). Towards a Virtual Archaeology. *CAA90. Computer Applications and Quantitative Methods in Archaeology 1990*, 132–139. Consulta: 10 de Mayo de 2022.

[https://publikationen.uni-](https://publikationen.uni-tuebingen.de/xmlui/bitstream/handle/10900/61199/21_Reilly_CAA_1990.pdf?sequence=2)

[tuebingen.de/xmlui/bitstream/handle/10900/61199/21_Reilly_CAA_1990.pdf?sequence=2](https://publikationen.uni-tuebingen.de/xmlui/bitstream/handle/10900/61199/21_Reilly_CAA_1990.pdf?sequence=2)

Renfrew, C., & Bahn, P. (2007). *Arqueología. Teorías, Métodos y Práctica* (3ra edición). Madrid.

Rozanek, M., Kudrna, P., Hozman, J., & Kopecky, P. (2013). Software monitor of vital functions for a mannequin realized on tablet PC. *2013 E-Health and Bioengineering Conference, EHB 2013*, 4–7. Consulta: 10 de Mayo de 2022.

<https://doi.org/10.1109/EHB.2013.6707318>

Rubin, J., & Chisnel, D. (2008). "What Makes Something Usable?" *Handbook of Usability Testing: How to Plan, Design And Conduct Effective Tests* (2nd Edition). 2014. Wiley, pp. 3-20. Consulta: 10 de Mayo de 2022.

https://media.wiley.com/product_data/excerpt/81/04701854/0470185481.pdf

Rucabado, J. (2010). *Aprendiendo a excavar: de la realidad a la simulación*. Consulta: 23 de Mayo de 2022.

<https://www.aidu-asociacion.org/wp-content/uploads/2019/12/TC-COM-195.pdf>

- Rumbaugh, J., I. Jacobson y G. Booch (2005). *The Unified Modeling Language Reference Manual*. Segunda edición. Reading: Addison-Wesley. Consulta: 2 de Mayo de 2022.
https://people.ucalgary.ca/~far/Lectures/SENG609-23/PDF/uml_refman.pdf
- Ruusuvuori, P., Nykter, M., M??kiraatikka, E., Lehmussola, A., Korpelainen, T., Erkkil??, T., & Yli-Harja, O. (2007). Microarray simulator as educational tool. *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 5919–5922. Consulta: 10 de Mayo de 2022.
<https://doi.org/10.1109/IEMBS.2007.4353695>
- Salama, M., & Shawish, A. (2013). Junosphere: Towards professional networks education. *Proceedings - International Computer Software and Applications Conference*, (figure 3), 290–291. Consulta: 10 de Mayo de 2022.
<https://doi.org/10.1109/COMPSAC.2013.53>
- Santos, R., Chibante, R., Araújo, A., & Carvalho, A. (2009). Educational Tool for Teaching of Semiconductors Keywords. *Semiconductors*, 1–9. Consulta: 10 de Mayo de 2022.
<https://ieeexplore.ieee.org/document/5279266>
- Seven Summits Studio. (2015). Avion Flight Simulator 2015. Consulta: 12 de Mayo de 2022.
<https://www.microsoft.com/en-us/p/avion-flight-simulator-2015/9nblggh1p8lq>
- Shaikh, O., & Shahzad, F. (2017). Visual-netsim: Development of an interactive browser based network simulator. *FTC 2016 - Proceedings of Future Technologies Conference*, (December), 718–724. Consulta: 12 de Mayo de 2022.
<https://doi.org/10.1109/FTC.2016.7821684>

- Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3–4), 591–611. Consulta: 12 de Mayo de 2022.
<https://doi.org/10.2307/2333709>
- Smithsonian National Museum of Natural History. (n.d.). Teaching Activities. Retrieved May 20, 2018. Consulta: 12 de Mayo de 2022.
https://web.archive.org/web/20181128181933/https://anthropology.si.edu/outreach/Teaching_Activities/index.htm
- Soares, L. R., & Casanova Alcalde, V. H. (2006). An educational robotic workstation based on the Rhino XR4 robot. *Proceedings - Frontiers in Education Conference, FIE*, 7–12. Consulta: 12 de Mayo de 2022.
<https://doi.org/10.1109/FIE.2006.322329>
- Society for American Archaeology. (n.d.-a). Archaeology for Educators. Consulta: 12 de Mayo de 2022.
https://web.archive.org/web/20181225111432/http://www.saa.org/publicftp/PUBLI/C/educators/01_introduction.html
- Society for American Archaeology. (n.d.-b). What is archaeology? Consulta: 12 de Mayo de 2022.
https://web.archive.org/web/20181225111643/http://www.saa.org/publicftp/public/educators/03_what.html
- Stamatescu, I., Stamatescu, G., Fagarasan, I., Arghira, N., Calofir, V., & Iliescu, S. S. (2017). ASID: Advanced System for Process Control towards Intelligent Specialization in the Power Engineering Field. *Proceedings of the 2017 9Th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing*

Systems: Technology and Applications (Ideaacs), Vol 1, (September), 475–480.

Consulta: 12 de Mayo de 2022.

<https://ieeexplore.ieee.org/document/8095126>

Stroustrup, B. (2013). *The C ++ Programming* (4th Edition). Consulta: 12 de Mayo de 2022.

<https://dl.acm.org/doi/book/10.5555/2543987>

Tachikawa, Y., & Nakamura, T. (2013). PMBOK simulator for acquiring decision making competencies in project management education. *Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering, TALE 2013*, (August), 41–46. Consulta: 12 de Mayo de 2022.

<https://doi.org/10.1109/TALE.2013.6654396>

Truchly, P., Danielovič, L., Dukát, M., & Pôbiš, A. (2016). Educational and simulation portal for Internet protocol version 6. *ICETA 2015 - 13th IEEE International Conference on Emerging ELearning Technologies and Applications, Proceedings*. Consulta: 12 de Mayo de 2022.

<https://doi.org/10.1109/ICETA.2015.7558520>

Ur Rehman Khan, A., Bilal, S. M., & Othman, M. (2013). A performance comparison of open source network simulators for wireless networks. *Proceedings - 2012 IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2012*, 34–38. Consulta: 12 de Mayo de 2022.

<https://doi.org/10.1109/ICCSCE.2012.6487111>

Vega-Medina, L., Perez-Gutierrez, B., Tibamoso, G., Uribe-Quevedo, A., & Jaimes, N. (2014). VR central venous access simulation system for newborns. *Proceedings - IEEE Virtual Reality*, 121–122. Consulta: 12 de Mayo de 2022.

<https://doi.org/10.1109/VR.2014.6802081>

Wang, M., Sun, L., Du, Z., & Jia, Z. (2006). The design and implementation of virtual system for the robot-assisted setting-bone surgery. *2nd ASM-IEEE EMBS Conference on Bio-, Micro- and Nanosystems*, 53–57. Consulta: 12 de Mayo de 2022.

<https://doi.org/10.1109/BMN.2006.330928>

Zbick, J. (2017). A web-based reference architecture for mobile learning: Its quality aspects and evaluation. *Proceedings - 2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017: Side Track Proceedings*, 230–235. Consulta: 12 de Mayo de 2022.

<https://doi.org/10.1109/ICSAW.2017.8>

