



**UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA**

**Doctorado en Ciencias Informáticas**

Tesis Presentada para Obtener el Grado de Doctor en Ciencias Informáticas

**Propuesta Metodológica y Arquitectónica para el Desarrollo de Juegos  
Serios Orientados a la Formación de Competencias Tecnológicas**

Autor  
Mag. Juan Carlos Sandí Delgado

Directora  
Dra. Patricia Alejandra Bazán

La Plata - Buenos Aires, Argentina.  
Mayo, 2023

## Dedicatoria

En primera instancia deseo dedicar este trabajo de tesis a Dios, por permitirme alcanzar este momento, guiar me y darme la salud física, emocional y mental para lograr cumplir con este objetivo.

A mi madre Obdulia Delgado Salazar y mi padre Calos Bolívar Sandí Bermúdez<sup>□</sup>, que me educaron, aconsejaron, inculcaron valores y me impulsaron a seguir siempre hacia adelante, con la frente muy en alto ante las dificultades y retos de cada día.

A mi hermana Ana Jenny Sandí Delgado, a mis hermanos Norman Sandí Delgado<sup>□</sup>, Luis Ángel Sandí Delgado, Eddy Sandí Delgado, sobrinos/as, que siempre me han apoyado y sé, que sienten mucho orgullo de mi progreso personal, académico y profesional.

A mis amigos/as de Costa Rica, Argentina, Uruguay, Colombia, Venezuela y Ecuador, por siempre estar presentes en todo momento.

A mi compañera y amiga, Zoila Siles Navarro, por todo su apoyo y soporte brindado.

A mis estudiantes de la Carrera de Informática Empresarial, en especial a Isaac Castro Guzmán y Steven Rojas Valverde, por todo el apoyo y asesoría.

A mis compañeros de trabajo de la Universidad de Costa Rica, en especial a, Rosa Julia Cerdas González, Maynor Jiménez Castro, Emilia Elizondo Murillo, Jane Segleau Earle, Cristian Brenes Granados, Ivannia Víquez Barrantes, Patricia Picado Chavaría, Michael Barquero, Steven Cruz Sancho, Mainor Cruz Alvarado, Ariel Arguedas Ulloa y Roberto Mesén Hidalgo, todos por estar siempre dispuestos a ayudarme, escucharme y convertirse en mis consejeros durante este período y, principalmente, por su apoyo y respaldo incondicional en todo momento.

## Agradecimientos

Agradezco la confianza y el apoyo de mis padres, hermanos y amigos, porque han contribuido positivamente para llevar a cabo esta difícil jornada.

Al personal académico del Doctorado en Ciencias Informáticas, quienes me asesoraban y guiaban durante el proceso formativo, y que con sus valiosos aportes me ayudaron a crecer como persona y como profesional.

Un especial agradecimiento a mi Directora de Tesis Dra. Patricia Alejandra Bazán, Docente e Investigadora de la Universidad Nacional de La Plata (UNLP) en Argentina, quien me acompañó, asesoró, guio y, principalmente, me respaldó en todo momento de la investigación realizada. Además, por ser una persona comprensiva, trabajadora y, sobre todo, con un alto don de humanismo.

A la Universidad de Costa Rica, en específico a la Sede del Atlántico y al profesorado por el permiso y la confianza otorgada, asimismo, un profundo agradecimiento a la Oficina de Asuntos Internacionales y Cooperación Externa (OAICE) por financiar este postgrado, y con ello, permitir el incremento de mis conocimientos para mejorar los procesos formativos y de investigación. A su vez, por potenciar la formación de profesionales más críticos y, sobre todo, más productivos para todo Costa Rica.

A la Facultad de Informática de la Universidad Nacional de La Plata (UNLP) y, en especial, a todas las personas que trabajan en la secretaría de postgrado de dicha facultad, por su cariño, apoyo y acompañamiento constante, facilitado durante el proceso académico realizado.

A mi gestora de becas en la OAICE, Vivian Madrigal Roldán, por su apoyo incondicional, motivación y respaldo brindado en todo momento, principalmente, cuando más lo requería por mi estado de salud, gracias.

## Índice de Contenidos

<b>Dedicatoria</b>	<b>i</b>
<b>Agradecimientos</b>	<b>ii</b>
<b>Índice de Contenidos</b>	<b>iii</b>
<b>Índice de Tablas</b>	<b>vii</b>
<b>Índice de Figuras</b>	<b>ix</b>
<b>Resumen</b>	<b>ix</b>
<b>Abstract</b>	<b>x</b>
<b>CAPÍTULO I. INTRODUCCIÓN</b>	<b>1</b>
<b>1.1. Motivación</b>	<b>2</b>
<b>1.2. Objetivos</b>	<b>4</b>
<b>1.3. Metodología de la investigación</b>	<b>4</b>
1.3.1. Selección de referencias bibliográficas	5
1.3.2. Preguntas de investigación	5
1.3.3. Estrategia de búsqueda	6
1.3.4. Palabras claves y cadenas de búsqueda	7
1.3.5. Criterios de inclusión/exclusión	8
1.3.6. Proceso preliminar	9
1.3.7. Proceso de selección final	9
1.3.8. Estudio de caso	9
<b>1.4. Aportes de la tesis</b>	<b>10</b>
<b>1.5. Estructura de la tesis</b>	<b>10</b>
<b>CAPÍTULO II. TIC y COMPETENCIAS TECNOLÓGICAS</b>	<b>13</b>
<b>2.1. Introducción</b>	<b>14</b>
<b>2.2. Conceptualización TIC, competencias y competencias tecnológicas</b>	<b>14</b>
2.3.1. Definición y caracterización de las TIC	14
2.3.2. Definición de competencias	16
2.3.3. Definición de competencias tecnológicas	17
<b>2.4. Importancia del desarrollo de competencias o habilidades tecnológicas en educación</b>	<b>20</b>
<b>2.5. Conclusiones del capítulo</b>	<b>21</b>
<b>CAPÍTULO III. JUEGOS SERIOS</b>	<b>22</b>
<b>3.1. Introducción</b>	<b>23</b>
<b>3.2. Conceptualización de juegos serios</b>	<b>23</b>
<b>3.3. Características de juegos serios</b>	<b>26</b>
<b>3.4. Antecedentes de uso de juegos serios en procesos de desarrollo de competencias TIC</b>	<b>29</b>
<b>3.5. Conclusiones del capítulo</b>	<b>33</b>

<b>CAPÍTULO IV. CLOUD COMPUTING</b>	<b>34</b>
4.1.    Introducción	35
4.2.    Conceptualización y características de <i>cloud computing</i>	35
4.2.1.    Características esenciales de <i>cloud computing</i>	38
4.3.    Modelos de servicio en <i>cloud computing</i>	39
4.4.    Modelos de despliegue en <i>cloud computing</i>	42
4.5.    Posibilidades de <i>cloud computing</i> en educación	45
4.6.    Conclusiones del capítulo	45
<b>CAPÍTULO V. CONSIDERACIONES TEÓRICAS EN EL DESARROLLO DE SOFTWARE</b>	<b>46</b>
5.1.    Introducción	47
5.2.    Lenguajes de programación	48
5.2.1.    Paradigmas de programación	48
5.2.2.    Tipos de lenguajes de programación	49
5.3.    Procesos de software	49
5.3.1.    Modelos de desarrollo de software	50
5.3.2.    Diseño Centrado en el Usuario	52
5.4.    Ingeniería de requerimientos	53
5.5.    Diseño del software	54
5.5.1.    Decisiones del diseño arquitectónico	54
5.5.2.    Estilos arquitectónicos	56
5.5.3.    Diagramas de flujo de datos	58
5.6.    Codificación del software	58
5.6.1.    Características del código del software	59
5.6.2.    Reutilización de software	60
5.7.    Pruebas de software	61
5.7.1.    Técnicas de pruebas de software	61
5.8.    Conclusiones del capítulo	62
<b>CAPÍTULO VI. REVISIÓN DE METODOLOGÍAS UTILIZADAS PARA EL DISEÑO DE JUEGOS SERIOS</b>	<b>64</b>
6.1.    Introducción	65
6.2.    Definición de los criterios de análisis	65
6.3.    Recopilación de las metodologías	70
6.3.1.    Metodología EMERGO	70
6.3.2.    Metodología EDoS	80
6.3.3.    Metodología LEGADEE	84
6.3.4.    Metodología SAVIE	87
6.3.5.    Modelo DODDEL	90
6.3.6.    Metodología VGSCL	93

6.3.7. Metodología MECONESIS	95
6.3.8. Metodología MPJu+a	100
6.3.9. Metodología MPDSG	103
<b>6.4. Análisis de resultados según criterios de evaluación</b>	<b>105</b>
6.4.1. Aspectos generales	105
6.4.2. Aspectos de diseño	107
6.4.3. Aspectos metodológico-pedagógicos	111
6.4.4. Aspectos de análisis	112
<b>6.5. Conclusiones del capítulo</b>	<b>114</b>
<b>CAPÍTULO VII. REVISIÓN DE ARQUITECTURAS UTILIZADAS PARA EL DISEÑO DE VIDEOJUEGOS</b>	<b>117</b>
<b>7.1. Introducción</b>	<b>118</b>
<b>7.2. Definición de los criterios de análisis</b>	<b>119</b>
<b>7.3. Recopilación de investigaciones relacionadas con propuestas de arquitecturas para el diseño o desarrollo de videojuegos</b>	<b>122</b>
7.3.1. Propuesta de (Maggiorini et al., 2016)	123
7.3.2. Propuesta de (El Mawas, 2014)	128
7.3.3. Propuesta de (Gutiérrez-Hernández et al., 2013)	132
7.3.4. Propuesta de (Söbke & Streicher, 2016)	138
7.3.5. Propuesta de (Stavrev et al., 2018)	144
7.3.6. Propuesta de (Flores et al., 2019)	148
7.3.7. Propuesta de (Mizutani et al., 2021)	153
7.3.8. Propuesta de (Ollsson et al., 2015)	156
7.3.9. Propuesta de (Carvalho, Bellotti, Hu, et al., 2015)	160
7.3.10. Propuesta de (Scacchi, 2017)	164
7.3.11. Propuesta de (Yessad et al., 2010)	166
7.3.12. Propuesta de (Costa et al., 2016)	168
7.3.13. Propuesta de (Ismail & Belkhouche, 2019)	173
<b>7.4. Análisis de resultados según criterios de evaluación</b>	<b>176</b>
7.4.1. Aspectos generales	176
7.4.2. Aspectos arquitectónicos	178
7.4.3. Aspectos pedagógicos	180
7.4.4. Aspectos de análisis	183
<b>7.5. Conclusiones del capítulo</b>	<b>183</b>
<b>CAPÍTULO VIII. PROPUESTA METODOLÓGICA Y ARQUITECTÓNICA PARA EL DISEÑO DE JUEGOS SERIOS</b>	<b>185</b>
<b>8.1. Introducción</b>	<b>186</b>
<b>8.2. Pasos metodológicos y arquitectónicos para el diseño de juegos serios</b>	<b>186</b>
<b>8.3. Propuesta metodológica y arquitectónica INTEGRA+506</b>	<b>187</b>
8.3.1. Fase 1. Consideraciones generales	188
8.3.2. Fase 2. Lenguaje de programación	189
8.3.3. Fase 3. Procesos de software	190

8.3.4	Fase 4. Ingeniería de requerimientos	191
8.3.5	Fase 5. Diseño de software – Aspectos arquitectónicos	192
8.3.6	Fase 6. Codificación del software	196
8.3.7	Fase 7. Validación de la metodología	197
<b>8.4</b>	<b>Conclusiones del capítulo</b>	<b>197</b>
<b>CAPÍTULO IX. PROTOTIPO NO FUNCIONAL DE SKILLNET</b>		<b>199</b>
<b>9.1.</b>	<b>Introducción</b>	<b>200</b>
<b>9.2.</b>	<b>Implementación de INTEGRA+506 + SkillNet</b>	<b>200</b>
9.2.1.	Fase 1. Consideraciones generales	201
9.2.2.	Fase 2. Lenguaje de programación	204
9.2.3.	Fase 3. Procesos de software	205
9.2.4.	Fase 4. Ingeniería de requerimientos	206
9.2.4.1	Requerimientos funcionales	206
9.2.4.2	Requerimientos no funcionales	216
9.2.5.	Fase 5. Diseño de software – aspectos arquitectónicos	231
9.2.6.	Fase 6. Codificación del software	233
9.2.7.	Fase 7. Validación de la metodología	234
<b>9.3.</b>	<b>Interfaces del prototipo SkillNet</b>	<b>235</b>
<b>9.4.</b>	<b>Conclusiones del capítulo</b>	<b>252</b>
<b>CAPÍTULO X. CONCLUSIONES Y LÍNEA DE TRABAJOS FUTUROS</b>		<b>253</b>
<b>10.1.</b>	<b>Introducción</b>	<b>254</b>
<b>10.2.</b>	<b>Conclusiones de la tesis</b>	<b>254</b>
<b>10.3.</b>	<b>Líneas de trabajo futuro</b>	<b>257</b>
<b>10.4.</b>	<b>Producción científica</b>	<b>258</b>
<b>ANEXOS</b>		<b>260</b>
<b>REFERENCIAS</b>		<b>261</b>

## Índice de Tablas

### **Capítulo 1**

<b>Tabla 1.2</b> Definición de cadenas de búsqueda según tema y palabras clave	8
--	---

### **Capítulo 2**

<b>Tabla 2.1.</b> Definiciones del Término TIC	15
<b>Tabla 2.2.</b> Definiciones del término competencias	16
<b>Tabla 2.3.</b> Definiciones del término competencias tecnológicas	19

### **Capítulo 3**

<b>Tabla 3.1</b> Juegos serios utilizados en procesos de desarrollo de competencias	32
---	----

### **Capítulo 6**

<b>Tabla 6.1</b> <i>Criterios de análisis de las metodologías para el diseño de juego serios</i>	67
<b>Tabla 6.2</b> <i>Metodologías seleccionadas para análisis</i>	70
<b>Tabla 6.3</b> Preguntas para la fase de análisis	75
<b>Tabla 6.4</b> Componentes de entrada de datos y su función componente	76
<b>Tabla 6.5</b> Componentes de entrada de datos y su función	79
<b>Tabla 6.6</b> Resumen - Aspectos de análisis de EMERGO	79
<b>Tabla 6.7</b> Resumen - Aspectos de análisis de EDoS	84
<b>Tabla 6.8</b> Resumen - Aspectos de análisis de LEGADEE	87
<b>Tabla 6.9</b> Resumen - Aspectos de análisis de SAVIE	90
<b>Tabla 6.10</b> Resumen - Aspectos de análisis de DODDEL	92
<b>Tabla 6.11</b> Resumen - Aspectos de análisis de VGSCL	94
<b>Tabla 6.12</b> Resumen - Aspectos de análisis de MECONESIS	100
<b>Tabla 6.13</b> Resumen - Aspectos de análisis de MPJu+ta	103
<b>Tabla 6.14</b> Resumen - Aspectos de análisis de MPDSG	105
<b>Tabla 6.15</b> Criterio - Nivel educativo según metodologías analizadas	106
<b>Tabla 6.16</b> Criterio - Fundamentación según metodologías analizadas	107
<b>Tabla 6.17</b> Criterio - Definición de roles según metodologías analizadas	109
<b>Tabla 6.18</b> Criterio - Destinatarios	112
<b>Tabla 6.19</b> Criterios - Validación de objetivos, evaluación UX y resultados cuantificables	113

### **Capítulo 7**

<b>Tabla 7.1</b> Criterios definidos para el análisis de las propuestas arquitectónicas	120
<b>Tabla 7.2</b> Referencias seleccionadas para el análisis	123
<b>Tabla 7.3</b> Resumen - Aspectos de análisis de (Maggiorini et al., 2016).	128
<b>Tabla 7.4</b> Comparativa de diferentes metodologías de diseño para juegos serios	130
<b>Tabla 7.5</b> Resumen - Aspectos de análisis de (El Mawas, 2014)	132
<b>Tabla 7.6</b> Facetas de diseño del juego serio	134
<b>Tabla 7.7</b> Relación del personal involucrado en la arquitectura	136
<b>Tabla 7.8</b> Representación de los niveles de la arquitectura para el caso de estudio	137
<b>Tabla 7.9</b> Resumen - Aspectos de análisis de (Gutiérrez-Hernández et al., 2013)	138

<b>Tabla 7.10 Resumen - Aspectos de análisis de (Söbke &amp; Streicher, 2016)</b>	143
<b>Tabla 7.11 Resumen - Aspectos de análisis de (Stavrev et al., 2018)</b>	148
<b>Tabla 7.12 Clasificación de los juegos serios orientados a resultados</b>	149
<b>Tabla 7.13 Fases de progresión del juego serio basado en la arquitectura propuesta</b>	152
<b>Tabla 7.14 Resumen - Aspectos de análisis de (Flores et al., 2019)</b>	152
<b>Tabla 7.15 Resumen - Aspectos de análisis de (Mizutani et al., 2021)</b>	156
<b>Tabla 7.16 Refactorizaciones en T.WT.P.B</b>	158
<b>Tabla 7.17 Clasificación del proyecto final y la arquitectura de la UI</b>	159
<b>Tabla 7.18 Resumen - Aspectos de análisis de (Olsson et al., 2015)</b>	160
<b>Tabla 7.19 Resumen - Aspectos de análisis de (Carvalho, Bellotti, Hu, et al., 2015)</b>	163
<b>Tabla 7.20 Resumen - Aspectos de análisis de (Carvalho et al., 2015)</b>	163
<b>Tabla 7.21 Tecnologías para la ingeniería de software de juegos para computadora</b>	165
<b>Tabla 7.22 Resumen - Aspectos de análisis de (Scacchi, 2017)</b>	166
<b>Tabla 7.23 Resumen - Aspectos de análisis de (Yessad et al., 2010)</b>	168
<b>Tabla 7.24 Resumen - Aspectos de análisis de (Costa et al., 2016)</b>	173
<b>Tabla 7.25 Funciones del sistema de aprendizaje personalizado</b>	175
<b>Tabla 7.26 Mapeo de sistemas de software de aprendizaje personalizado</b>	175
<b>Tabla 7.27 Resumen - Aspectos de análisis de (Ismail &amp; Belkhouche, 2019)</b>	176
<b>Tabla 7.28 Criterio - País donde se desarrolla la propuesta arquitectónica</b>	177
<b>Tabla 7.29 Criterio - Nivel educativo según propuestas arquitectónicas</b>	178
<b>Tabla 7.30 Criterio - Fundamentación de la arquitectura</b>	179
<b>Tabla 7.31 Criterio - Tipo de dispositivo al que está orientada la arquitectura</b>	179
<b>Tabla 7.32 Criterio - Destinatarios de la arquitectura</b>	181
<b>Tabla 7.33 Criterio - Propósito o intención de la arquitectura</b>	182
<b>Tabla 7.34 Criterio - Nivel de complejidad</b>	182
<b>Tabla 7.35 Criterio - Tipo de producto por desarrollar</b>	183

## Capítulo 9

<b>Tabla 9. 1 Fases y pasos metodológicos y arquitectónicos de INTEGRA+506</b>	200
<b>Tabla 9. 2 Competencias del profesorado en TIC en España</b>	202
<b>Tabla 9. 3 Competencias e indicadores a potenciar en el profesorado</b>	203
<b>Tabla 9. 4 Módulo de Registro RF-1</b>	207
<b>Tabla 9. 5 Módulo Autentificación RF-2</b>	208
<b>Tabla 9. 6 Módulo Contraseña RF-3</b>	208
<b>Tabla 9. 7 Módulo de Menú Inicio del Juego RF-4</b>	209
<b>Tabla 9. 8 Módulo Etapa 1 RF-5</b>	210
<b>Tabla 9. 9 Módulo Etapa #2 RF-6</b>	213
<b>Tabla 9. 10 Módulo Etapa #3 RF-7</b>	215
<b>Tabla 9. 11 Especificación de Requerimientos No Funcionales y comunes RNF</b>	216
<b>Tabla 9. 12 Caso de Uso Login</b>	217
<b>Tabla 9. 13 Caso de uso solicitar recuperación de contraseña</b>	218
<b>Tabla 9. 14 Caso de uso recuperación de contraseña</b>	219
<b>Tabla 9. 15 Caso de uso registro de usuario</b>	220
<b>Tabla 9. 16 Caso de uso etapa 1 - nivel 1 al 5 del juego</b>	220
<b>Tabla 9. 17 Caso de uso etapa 2 - nivel 1 al 2 del juego</b>	222
<b>Tabla 9. 18 Caso de uso etapa 3 – nivel 1</b>	223
<b>Tabla 9. 19 Caso de uso agregar usuario al grupo</b>	224
<b>Tabla 9. 20 Caso de uso eliminar grupo</b>	225
<b>Tabla 9. 21 Caso de uso agregar usuario al grupo</b>	226

<b>Tabla 9. 22</b> Caso de uso eliminar usuario al grupo	226
<b>Tabla 9. 23</b> Reglas de negocio	227

## Índice de Figuras

### Capítulo 4

<b>Figura 4.1</b> Integración de los modelos de cloud computing	44
<b>Figura 4.2</b> Modelo de nube integrado propuesto por NIST	44

### Capítulo 5

<b>Figura 5.1</b> Modelo del diseño de procesos centrados en el usuario.	52
<b>Figura 5.2</b> Interrogantes a considerar en el diseño arquitectónico	55
<b>Figura 5.3</b> Reutilización de software	60

### Capítulo 6

<b>Figura 6.1</b> Metodología para el desarrollo de casos EMERGO	71
<b>Figura 6.2</b> Componentes de entrada de datos para el desarrollo de casos	77
<b>Figura 6.3</b> Componentes de entrada de datos para entrega de casos	78
<b>Figura 6.4</b> Pestaña para la definición de objetivos pedagógicos en EDoS	81
<b>Figura 6.5</b> Proceso de 7 pasos para el diseño de SG, LEGADEE	86
<b>Figura 6.6</b> Modelo DODDEL	92
<b>Figura 6.7</b> Proceso de análisis para evaluar la experiencia del usuario - MECONESIS	97
<b>Figura 6.8</b> Modelo de escenario de MECONESIS	98
<b>Figura 6.9</b> Metodología MECONESIS	99
<b>Figura 6.10</b> Metodología MPJu+a: integración de la Usabilidad y Accesibilidad	102
<b>Figura 6.11</b> Modelo de proceso MPDSG	104
<b>Figura 6.12</b> Criterio - País de origen según metodologías analizadas	106
<b>Figura 6.13</b> Criterio - Definición de requerimientos según metodologías analizadas	108
<b>Figura 6.14</b> Criterio - Patrones de diseño	110
<b>Figura 6.15</b> Criterio - Reutilización de software	110
<b>Figura 6.16</b> Criterio – Diseño de prototipos	111

### Capítulo 7

<b>Figura 7.1</b> Resumen de una arquitectura estándar para motor de juego	125
<b>Figura 7.2</b> Diagrama de la arquitectura SMASH	126
<b>Figura 7.3</b> Diagrama de la arquitectura SMASH	127
<b>Figura 7.4</b> Dependencia entre los componentes de la arquitectura	130
<b>Figura 7.5</b> Arquitectura propuesta para juegos serios multiculturales	135
<b>Figura 7.6</b> Implementación de la arquitectura en el videojuego de temperaturas	137
<b>Figura 7.7</b> Modelo de la vista 4+1 de una arquitectura por Krutchén	139
<b>Figura 7.8</b> Vista de desarrollo de la arquitectura LernShooter	140
<b>Figura 7.9</b> Vista física del juego para aplicación web Energetika	141
<b>Figura 7.10</b> Framework para juegos serios adaptables para la salud	142
<b>Figura 7.11</b> Vista general de DiAS	145
<b>Figura 7.12</b> Arquitectura general de un juego serio	149
<b>Figura 7.13</b> Arquitectura propuesta para un juego serio basado en retos	150

<b>Figura 7.14</b> Interrelación de los módulos de la arquitectura propuesta	151
<b>Figura 7.15</b> Diagrama de secuencia de la arquitectura propuesta	151
<b>Figura 7.16</b> Frecuencia de estudios seleccionados con cada requisito de diseño de SW	153
<b>Figura 7.17</b> Estudios seleccionados con diferentes prácticas arquitectónicas	154
<b>Figura 7.18</b> Frecuencia entre mecánica del juego y prácticas arquitectónicas	155
<b>Figura 7.19</b> Diagrama de clases sintetizado para TimeBreaker y Frontilne	157
<b>Figura 7.20</b> Diagrama de clases T.W.T.P.B	157
<b>Figura 7.21</b> Diagrama de clases sintetizado de Hero y Gears of Love	158
<b>Figura 7.22</b> Actividades por acciones con herramientas y objetivos específicos	161
<b>Figura 7.23</b> Arquitectura SeGAE	167
<b>Figura 7.24</b> Procesos de validación identificados en las investigaciones	169
<b>Figura 7.25</b> Ajustes de los servicios brindados por las plataformas o portales	170
<b>Figura 7.26</b> Arquitectura general propuesta por (Costa et al., 2016)	172
<b>Figura 7.27</b> Criterio - Reutilización de componentes	180

## Capítulo 8

<b>Figura 8. 1</b> Secuencia metodológica para el diseño de juegos serios	198
---	-----

## Capítulo 9

<b>Figura 9. 1</b> Actores participantes en el diseño del prototipo de SkillNet	201
<b>Figura 9. 2</b> Diagrama de clases	232
<b>Figura 9. 3</b> Diagrama entidad relación	233
<b>Figura 9. 4</b> Página de inicio de SkillNet	235
<b>Figura 9. 5</b> Registro a SkillNet	235
<b>Figura 9. 6</b> Ingreso a SkillNet	236
<b>Figura 9. 7</b> Pantalla principal del juego en general	237
<b>Figura 9. 8</b> Estadísticas a nivel individual en SkillNet	238
<b>Figura 9. 9</b> Administración de grupos en SkillNet	239
<b>Figura 9. 10</b> Estadísticas a nivel grupal	239
<b>Figura 9. 11</b> Información teórica del contenido de la etapa 1 nivel 1	240
<b>Figura 9. 12</b> Ventana informativa ejemplos prácticos de la etapa 1 – nivel 1	241
<b>Figura 9. 13</b> Ejercicio práctico de la etapa 1 - Nivel 1	241
<b>Figura 9. 14</b> Mensaje de felicitaciones por completar correctamente el nivel 1	242
<b>Figura 9. 15</b> Mensaje informativo de desacuerdo en el nivel 1 – etapa 1	242
<b>Figura 9. 16</b> Mensajes informativo de avance de nivel y estrellas ganadas	243
<b>Figura 9. 17</b> Información teórica y práctica del contenido de la etapa 1 - nivel 2	244
<b>Figura 9. 18</b> Información teórica y práctica del contenido de la etapa 1 - nivel 3	245
<b>Figura 9. 19</b> Información teórica etapa 2 - nivel 1	246
<b>Figura 9. 20</b> Actividades etapa 2 - nivel 1 ejemplo A	246
<b>Figura 9. 21</b> Actividades etapa 2 - nivel 1 ejemplo B	247
<b>Figura 9. 22</b> Información etapa 2 - nivel 2	247
<b>Figura 9. 23</b> Ejercicios prácticos etapa 2 - nivel 2	248
<b>Figura 9. 24</b> Información general para la etapa 3	249
<b>Figura 9. 25</b> Información para la etapa 3 - nivel 1	250
<b>Figura 9. 26</b> Sistema de Ranking	251
<b>Figura 9. 27</b> Información de contacto	251



## Resumen

En la actualidad la formación de competencias tecnológicas en el profesorado se ha vuelto un factor imprescindible en el ámbito educativo. En este sentido los juegos serios han sido utilizados para entrenamiento, la formación, capacitación y adquisición de habilidades en diferentes áreas del conocimiento.

Si bien existen investigaciones que hacen referencia a la posibilidad que ofrecen los juegos serios para la formación de habilidades y competencias digitales, no se evidencian diseñados o arquitecturas de estos pensados exclusivamente para potenciar habilidades y competencias tecnológicas en el profesorado a nivel de educación superior.

Por tal razón y, en el marco del Doctorado en Ciencias Informáticas, se realizó una investigación con el objetivo de brindar una propuesta metodológica y arquitectónica que guíe paso a paso el diseño de juegos serios vinculados con la formación de habilidades o competencias tecnológicas. En consecuencia, se planteó una metodología y arquitectura de software para diseñar un prototipo no funcional del juego serio denominado SkillNet, el cual permite la formación de habilidades tecnológicas en el profesorado; específicamente las relacionadas con competencias y conocimientos básicos en redes de datos.

Para llevar adelante la investigación y cumplir con el objetivo propuesto, se realizó una indagación bajo un enfoque metodológico cualitativo y un diseño descriptivo, donde se realizó primeramente una revisión teórica acorde con los temas de investigación; principalmente los relacionados con metodologías y arquitecturas utilizadas para el diseño de juegos serios, así como las vinculadas con el diseño de prototipos. Posteriormente, se diseña y propone la metodología denominada INTEGRA+506 la cual a través de siete etapas y dieciséis pasos metodológicos guía el diseño de juegos serios orientados a la formación de competencias tecnológicas.

Seguido, tomando como referencia la metodología INTEGRA+506 se diseña un prototipo no funcional de un juego serio llamado SkillNet, el cual está orientado a formar habilidades y competencias en el área de las redes de datos. El prototipo de juego consta de tres etapas conformadas por diferentes niveles cada una, en las cuales el jugador debe enfrentar diversos desafíos para superarlos y desbloquear el siguiente nivel.

Finalmente, se concluye que la metodología **INTEGRA+506** mostró efectividad para guiar el trabajo realizado en el diseño del prototipo no funcional de **SkillNet**. Su aplicación ayudó a ordenar todo el proceso acorde con la validación realizada por el panel de expertos que revisó y analizó tanto la propuesta como el prototipo.

**Palabras Claves:** *Juegos serios; formación de competencias tecnológicas; prototipo; profesorado; propuesta metodológica y arquitectónica; INTEGRA+506; SkillNet; redes de datos.*

## Abstract

Nowadays, the formation of technological competencies in professors has become an essential factor in the educational field. In this sense, serious games have been used for training, education, training, and the acquisition of skills in different areas of knowledge.

Although there is research that refers to the possibility offered by serious games for the training of digital skills and competencies, there is no evidence of their design or architecture aimed exclusively at enhancing technological skills and competencies in teachers at the higher education level.

For this reason, and within the Ph.D. in Computer Science framework, research was carried out to provide a methodological and architectural proposal to guide the design of serious games linked to developing technological skills or competencies. Consequently, a methodology and software architecture was proposed to design a non-functional prototype of the serious game called SkillNet, which allows the formation of technological skills in teachers, specifically those related to competencies and basic knowledge in data networks.

In order to carry out the research and meet the proposed objective, an inquiry was conducted under a qualitative methodological approach and a descriptive design. In this inquiry, a theoretical review was first conducted according to the research topics, mainly those related to methodologies and architectures used to design serious games and those linked to the design of prototypes. Subsequently, the methodology called INTEGRA+506 is designed and proposed, which through seven stages and sixteen methodological steps, guides the design of serious games oriented to the formation of technological competencies.

Next, taking the INTEGRA+506 methodology as a reference, a non-functional prototype of a serious game called SkillNet is designed and oriented to train skills and competencies in data networks. The game prototype consists of three stages, each with different levels, in which the player must face different challenges to overcome them and unlock the next level.

Finally, it is concluded that the INTEGRA+506 methodology effectively guided the work done in the design of the non-functional SkillNet prototype. Its application helped order the whole process according to the validation performed by the panel of experts that reviewed and analyzed the proposal and the prototype.

**Keywords:** *Serious games; technological skills training; prototype; teaching staff; methodological and architectural proposal; INTEGRA+506; SkillNet; data networks.*

# CAPÍTULO I. INTRODUCCIÓN

## 1.1. Motivación

Ante el surgimiento de las Tecnologías de la Información y de la Comunicación (TIC) y su rápido avance e impacto a nivel educativo, las Instituciones de Educación Superior (IES) se ven obligadas a enfrentarse al reto de mantenerse actualizadas ante las nuevas demandas de la sociedad actual (Castells, 2009; Cuevas-Cordero & García-Fallas, 2014; Vega-Hernández et al., 2018). Asimismo, las TIC brindan a las IES el potencial para la creación de escenarios de aprendizaje que permiten el acceso inmediato tanto de información como de recursos tecnológicos que facilitan la formación permanente del estudiantado y el profesorado<sup>1</sup> en tecnologías digitales (Cabero, 2010; Cabero et al., 2015).

Aguaded & Cabero (2014), afirman que el principal problema que ha manifestado el profesorado es la actualización profesional, ya que requieren de diversas actividades de capacitación relacionadas a potenciar sus habilidades y competencias tecnológicas<sup>2</sup>, mismas requeridas para utilizar e integrar las TIC en los procesos formativos.

Diferentes investigadores (Aguaded & Cabero, 2014; Brown, 2005; Cuberos de Quintero & Vivas, 2017; Sandí-Delgado & Cordero-Badilla, 2013), indican que el uso que les brinda el personal docente a las TIC es aún escaso o superficial. Además, no se utiliza las TIC de manera eficaz a pesar de la disponibilidad de hardware y software existente (Chen, 2010).

En investigaciones realizadas por Teo (2009) y Scherer, Siddiq, & Teo (2015), se afirma que el proceso de enseñanza y aprendizaje apoyados por las TIC pueden ser innovados mediante la incorporación de juegos serios en las metodologías de trabajo. Lo anterior, debido a que los mismos pueden ser utilizados para la adquisición de habilidades, formación de competencias, cambios de actitud, comportamiento y, pueden potenciar la motivación y el rendimiento del aprendizaje (Boyle et al., 2011; Chang et al., 2012; Giannakos, 2013; Sung & Hwang, 2013).

En este sentido, se indica en varios estudios (Almerich et al., 2016; Bezanilla et al., 2014; Boyle et al., 2012; Calabor et al., 2017; Connolly et al., 2012; Del-Moral &

<sup>1</sup> Es importante aclarar que por razones culturales se utiliza en Costa Rica la palabra “docente” como sinónimo de “profesor” o “profesorado”. Por ende, en este documento se utilizan ambos términos como sinónimos.

<sup>2</sup> En esta investigación se utiliza “competencias tecnológicas” y “competencias digitales” como sinónimos para hacer referencias a las habilidades TIC (se amplía y justifica en detalle en la sección 2.2.3).

Guzmán-Duque, 2014; Guillén-Nieto & Aleson-Carbonell, 2012; Hainey et al., 2016; Kiili et al., 2018; Kwon & Lee, 2016; López, 2016; Marsh, 2011; Michael & Chen, 2006; Pontual et al., 2018; M. Qian & Clark, 2016; Romero et al., 2015; Romero & Turpo, 2012; Wrzesien & Alcañiz, 2010), que los juegos serios pueden adquirir un rol protagónico ante la posibilidad de convertirse en un medio para el acercamiento del profesorado a las tecnologías digitales.

Sandí & Bazán (2017) realizaron una investigación con el objetivo de plantear retos y desafíos para la ejecución de juegos serios educativos utilizando las posibilidades que brinda el modelo de SaaS en Cloud Computing. Como resultado del trabajo se obtuvo una lista de características funcionales que los juegos serios educativos adquirirían al ser utilizados como software como servicio.

En esta misma línea, Sandí & Sanz (2018a) llevaron a cabo una investigación con el propósito de indagar el estado del arte de los juegos serios educativos y sus posibilidades para la adquisición de competencias tecnológicas en la formación del profesorado. Como resultado se obtuvo la elaboración de una propuesta de integración de juegos serios para la formación de competencias tecnológicas en el profesorado y para la mejora de sus actitudes frente a estas tecnologías. Además, se elaboró de una propuesta para la evaluación del grado de aceptación de las tecnologías digitales por parte del profesorado, en particular sobre juegos serios.

Si bien existe variedad de investigaciones (Angelini et al., 2017; Chipia, 2011; Girard et al., 2013; Guenaga et al., 2015; Kwon & Lee, 2016; Muñoz et al., 2011; Prendes & Gutiérrez, 2013; Sawyer & Smith, 2008; Tsai et al., 2016; Wrzesien & Alcañiz, 2010; Yusoff et al., 2010), que hacen referencia a la posibilidad que ofrecen los juegos serios para la formación de habilidades y competencias digitales, no se evidencian juegos serios diseñados exclusivamente para potenciar habilidades y competencias tecnológicas del profesorado a nivel de educación superior. Asimismo, no se ha localizado o identificado en las referencias bibliográficas analizadas, la existencia de metodologías específicas que orienten o sirvan de guía para la creación y diseño de juegos serios orientados a la formación de competencias digitales.

Los resultados obtenidos en las investigaciones realizadas dan pie y son referente teórico para la elaboración de una investigación que permita proponer una

metodología y arquitectura de software que guíe los pasos a seguir durante la creación e implementación de juegos serios que vayan a ser utilizados para la formación de habilidades o competencias tecnológicas. En consecuencia, se plantea aplicar la metodología y la arquitectura de software que se proponga para crear y diseñar un prototipo no funcional de juego serio que pueda utilizarse para la formación de habilidades o competencias tecnológicas.

En adelante, el capítulo se organiza de la siguiente manera. La sección 1.2 corresponde a los objetivos de investigación. Seguido, la sección 1.3 presenta la metodología de investigación utilizada. Luego, la sección 1.4 reúne los aportes de la tesis. Finalmente, la sección 1.5 presenta la estructura de esta.

## 1.2. Objetivos

Para llevar adelante esta investigación se plantea un objetivo general y, luego, un conjunto de objetivos específicos, los cuales permiten viabilizar el objetivo general, mismos que se describen a continuación:

### a) Objetivo general

- ✓ Definir una metodología y arquitectura de software para el diseño de juegos serios que permitan la adquisición o formación de competencias tecnológicas.

### b) Objetivos específicos

- ✓ Estudiar los conceptos básicos vinculados con las TIC, competencias tecnológicas, juegos serios y desarrollo de software.
- ✓ Describir antecedentes de metodologías utilizadas en el diseño de software, en particular de juegos serios, así como las arquitecturas de software y de despliegue utilizadas.
- ✓ Elaborar una propuesta metodológica y arquitectónica para el diseño de juegos serios que potencien la adquisición de competencias tecnológicas.

- ✓ Validar la metodología a través de un panel de expertos o a través del diseño de un prototipo no funcional de juego serio, de manera que permita analizar el alcance del aporte de esta tesis y las posibles mejoras por realizar.
- ✓ Elaborar un informe con los resultados y las conclusiones a las que se arriba.

### 1.3. Metodología de la investigación

Para responder al objetivo de este estudio, se propone una investigación fundamentada en el diseño descriptivo, con un enfoque metodológico cualitativo (Hernández Sampieri et al., 2014), la cual está constituida en tres fases; en la **primera fase** se realiza una búsqueda, revisión y selección de referencias bibliográficas relacionadas a los temas y objetivos propuestos en esta investigación. Lo cual permite consolidar el marco referencial de la tesis que incluye los temas de competencias tecnológicas, juegos serios, etapas en el desarrollo de software, metodologías para el diseño de juegos serios, arquitectura de software, modelos de despliegue, entre otros. En la **segunda fase** se elabora la propuesta metodológica y arquitectónica que permite guiar paso a paso el proceso de planificación y desarrollo de Juegos Serios. Los pasos metodológicos propuestos, se formulan con base en la revisión de las diferentes referencias seleccionadas para tal propósito. En la **tercera fase**, se procede a someter la metodología y arquitectura propuesta al juicio de expertos para validar su funcionalidad para la creación de juegos serios que puedan ser utilizado para la formación de habilidades tecnológicas. En las siguientes secciones, se describe con detalle los pasos metodológicos de la primera fase citada.

#### 1.3.1. Selección de referencias bibliográficas

Para realizar la revisión y selección de las referencias bibliográficas, se toma como referencia los pasos metodológicos propuestos por Kitchenham (2004, 2007), Kitchenham et al. (2009, 2010) y Petersen et al. (2015), quienes recomiendan generar preguntas de investigación, definir estrategias de búsqueda en diferentes bases de datos de consulta para localizar artículos en revistas, libros o capítulos de los mismos, tesis, reportes de investigación y actas de congresos, definir palabras

claves, cadenas de búsqueda, definir criterios de inclusión y exclusión según el idioma, país de origen y área de interés, que permitan la selección final de las referencias bibliográficas (Cruz-Alvarado & Bazán, 2018; Sandí-Delgado & Bazán, 2017, 2019, 2020, 2021; Sandí-Delgado & Cruz-Alvarado, 2018; Sandí-Delgado & Sanz, 2018a).

### 1.3.2. Preguntas de investigación

Acorde al objetivo de estudio propuesto, se delimitaron trece (13) preguntas de investigación (PI) que contienen los criterios apropiados para realizar la inclusión o exclusión de las referencias bibliográficas.

- Plx1 ¿Cómo se define o describe el concepto de Tecnologías de la Información y la Comunicación (TIC)?
- Plx2 ¿Cómo se define el concepto de competencias o competencias tecnológicas?
- Plx3 ¿Cómo se definen actualmente los juegos serios?
- Plx4 ¿Cuáles son las principales características, requerimientos técnicos o funcionales de los juegos serios?
- Plx5 ¿Qué antecedentes existen de la utilización de juegos serios para la generación y desarrollo de competencias tecnológicas?
- Plx6 ¿Cómo se define o describe el concepto de *cloud computing*?
- Plx7 ¿Cuáles son las características esenciales de *cloud computing*?
- Plx8 ¿Cómo se describen los modelos de servicio y despliegue en *cloud computing*?

- Plx9 ¿Qué aspectos se deben considerar en el desarrollo de software?, ¿Cómo se describen o caracterizan estos aspectos?
- Plx10 ¿Cuáles metodologías se han utilizado para el diseño de juegos serios?, ¿qué antecedentes existen en este sentido?
- Plx11 ¿Qué aspectos metodológicos se consideran en la literatura para el diseño de juegos serios en la formación de competencias tecnológicas?
- Plx12 ¿Cuáles arquitecturas se han utilizado para el diseño de videojuegos o juegos serios?, ¿qué antecedentes existen en este sentido?
- Plx13 ¿Qué aspectos arquitectónicos se consideran en la literatura para el diseño de videojuegos o juegos serios?

### 1.3.3. Estrategia de búsqueda

Se utilizó como referencia a Martínez (2016), quien propone como estrategia de búsqueda; consultar y utilizar diferentes bases de datos científicas y académicas, tales como IEEE Xplore Digital Library<sup>3</sup>, ScienceDirect<sup>4</sup>, Springer<sup>5</sup>.

Además, se consideró la utilización del repositorios institucionales digitales reconocidos en los países donde se desenvuelven laboralmente los investigadores, tales como el SEDICI<sup>6</sup> (Servicio de Difusión de la Creación Intelectual de la Universidad nacional de La Plata (UNLP)) y el SIBDI<sup>7</sup> (Sistema de Bibliotecas, Documentación e Información de la Universidad de Costa Rica (UCR)), las cuales fueron elegidas porque facilitan el acceso a documentación publicada en revistas nacionales e internacionales, actas de los congresos con referato y reconocimiento por la comunidad científica internacional (Sandí-Delgado & Bazán, 2019; Sandí-Delgado & Sanz, 2018b).

<sup>3</sup> <http://ieeexplore.ieee.org/Xplore/home.jsp?reload=true>

<sup>4</sup> <http://www.sciencedirect.com/>

<sup>5</sup> <https://rd.springer.com/>

<sup>6</sup> <http://sedici.unlp.edu.ar/>

<sup>7</sup> <http://sibdi.ucr.ac.cr/>

#### 1.3.4. Palabras claves y cadenas de búsqueda

La investigación de las referencias bibliográficas se realizó para los períodos comprendidos entre el 2005 al 2020 inclusive. Para ello, la búsqueda se conformó a través de palabras claves en idioma español e inglés, tales como competencias tecnológicas - *technological skills*, competencias digitales - *digital competences*, tecnologías digitales - *digital technologies*, tecnologías de la información y la comunicación (TIC) - *information and communication technologies* (ICT), juegos serios - *serious games*, computación en la nube - *cloud computing*, arquitectura de software - *software architecture*, metodologías - *methodologies*, desarrollo de software - *software development*, características - *Characteristics*, desarrollo - *development*. Es importante indicar que se consideraron todas las palabras claves para el desarrollo de la investigación, sin embargo, no se utilizaron todas en la formación de las cadenas de búsqueda.

Las cadenas de búsqueda fueron compuestas por la combinación de algunas palabras claves en español e inglés, agrupadas por tema y palabras claves, tales como se pueden observar en la [Tabla 1.1](#).

**Tabla 1.1 Definición de cadenas de búsqueda según tema y palabras clave**

Tema	Palabras clave	Cadenas de búsqueda español e inglés
Competencias tecnológicas y TIC	Competencias tecnológicas, <i>Technological skills</i> , Competencias digitales, <i>Digital competences</i> ,	Competencias tecnológicas, <i>Technological skills</i>
		Competencias digitales, <i>Digital competences</i>
		Tecnologías de información y la comunicación, <i>information and communication technologies</i>
		TIC, ICT
Juegos serios	Juegos serios, <i>Serious games</i> , Características, <i>Characteristics</i> .	Características “juegos serios”, <i>Characteristics “serious games”</i>
		“Características de los juegos serios”, <i>“Serious games Characteristics”</i>
Desarrollo de software	<i>Software</i> , metodologías, <i>methodologies</i> , desarrollo, <i>development</i>	Metodologías de desarrollo de <i>software</i> , <i>Software development methodologies</i>
		Desarrollo de software, <i>Software development</i>

En la siguiente sección se describen los criterios para la inclusión y exclusión de referencias o documentos primarios.

### 1.3.5. Criterios de inclusión/exclusión

Para la **inclusión** de referencias bibliográficas se utilizaron los siguientes criterios de inclusión (CI):

- Clx1 Referencias en idioma español e inglés.
- Clx2 Referencias electrónicas de texto completo.
- Clx3 Documentos que definen el concepto de competencias o competencias tecnológicas.
- Clx4 Documentos que definen o describen el concepto de juegos serios.
- Clx5 Documentos que identifican características generales o funcionales de los juegos serios.
- Clx6 Documentos que definen o describen requerimientos técnicos de los juegos serios.
- Clx7 Documentos que describen antecedentes de utilización de juegos serios para la generación y desarrollo de competencias tecnológicas.
- Clx8 Documentos que definen o describen las etapas del desarrollo de software.
- Clx9 Documentos que definen o describen el concepto de juegos serios.

CIx10 Documentos que definen o describen metodologías específicas que permiten llevar adelante el desarrollo de juegos serios para la formación de competencias tecnológicas.

CIx11 Documentos que definen o describen arquitecturas y modelos de despliegue que permiten llevar adelante el desarrollo de juegos serios para la formación de competencias tecnológicas.

A. Para la **exclusión** de referencias bibliográficas se utilizaron los siguientes criterios de exclusión (CE):

CEx1 Referencias con idioma distinto al español e inglés.

CEx2 Referencias a las que no se logró tener acceso al texto completo.

CEx3 Referencias que no indicaban la autoría o fecha de publicación.

CEx4 Referencias publicadas en revistas sin referato internacional.

CEx5 Referencias con temas no relacionados con el objetivo de investigación.

### 1.3.6. Proceso preliminar

Se inició con un proceso de selección preliminar al ejecutarse las cadenas de búsqueda en las fuentes seleccionadas, luego se procedió a realizar una lectura del título, palabras claves y resumen de cada una de las referencias bibliográficas encontradas, a las cuales se les aplicó los criterios de inclusión y exclusión, con el propósito de generar la lista de referencias a considerar para una lectura completa y minuciosa.

En algunas ocasiones, una misma referencia facilitaba el acceso a otra bibliografía relacionada al tema en estudio, ya fuese del mismo o de diferentes autores, a las cuales se les aplicó por igual los criterios de inclusión o exclusión según fuese el caso.

### 1.3.7. Proceso de selección final

Una vez concluida la fase preliminar, se procedió con la fase final, en la cual se realizó lectura completa de las referencias seleccionadas a las cuales se les volvió a

aplicar los criterios de inclusión y exclusión para garantizar que estaban relacionados al objetivo de la investigación y, en caso de no estarlo, no fue considerado.

### **1.3.8. Estudio de caso**

El estudio de caso consiste someter la propuesta metodológica al juicio de expertos con el objetivo de validar la metodología y la arquitectura propuesta, de manera que permita analizar el alcance de los aportes de esta tesis y las posibles mejoras por realizar.

La metodología de investigación planteada favorece la recolección de información veraz y precisa para poder cumplir con los objetivos propuestos y apoyar los argumentos expresados por el investigador con respecto a la propuesta metodológica y arquitectónica para el desarrollo de juegos serios orientados a la formación de habilidades o competencias tecnológicas.

### **1.4. Aportes de la tesis**

El principal aporte que brinda esta investigación radica en la obtención de una propuesta metodológica y arquitectónica que permita orientar y guiar el proceso de diseño e implementación de prototipos de juegos serios, específicamente, los que hayan sido pensados para ser utilizados en la formación de habilidades o competencias tecnológicas.

Otro de los aportes importantes de la tesis consiste en que, a partir de la metodología desarrollada, permite identificar cual es la población meta más apta para utilizar el juego serio en la formación de habilidades o competencias tecnológicas.

### **1.5. Estructura de la tesis**

El presente trabajo de investigación da a conocer la propuesta metodológica y arquitectónica para el desarrollo de juegos serios orientados a la formación o capacitación en tecnologías digitales. También, reúne los resultados obtenidos a partir del juicio de expertos, a partir de la propuesta presentada con anterioridad. El trabajo está constituido por IX capítulos, cuyos contenidos son:

- ✓ **El capítulo I**, está conformado por la introducción, la motivación, el planteamiento del problema y justificación, objetivo general, objetivos específicos, la metodología de investigación, aportes y organización de la tesis.
- ✓ **El capítulo II**, corresponde a la primera parte de la revisión bibliográfica, en la cual se aborda el tema de las TIC y competencias tecnológicas. Específicamente, el capítulo se constituye por la introducción, seguido se aborda el tema de las TIC y competencias tecnológicas; conceptualización, competencias o habilidades tecnológicas. Por último, el impacto del desarrollo de competencias o habilidades tecnológicas.
- ✓ **El capítulo III**, corresponde a la segunda parte de la revisión bibliográfica, se aborda el tema de juegos serios. El capítulo está conformado por la introducción, conceptualización, características generales y funcionales de juegos serios. Finalmente, se abordan antecedentes de uso de juegos serios en procesos de desarrollo de competencias tecnológicas.
- ✓ **El capítulo IV**, corresponde a la tercera parte de la revisión bibliográfica, se aborda el tema de *cloud computing*. El capítulo inicia con la conceptualización y caracterización de *cloud computing*. Luego, se citan y describen tanto los modelos de servicios como los modelos de despliegue. Finalmente, se describen las posibilidades de *cloud computing* en educación.
- ✓ **El capítulo V**, corresponde a la cuarta parte de la revisión bibliográfica, el cual está relacionado con aspectos teóricos a considerar en el desarrollo de software. El capítulo se compone concretamente por la introducción, lenguajes de programación utilizados para el desarrollo de *software*. Seguido, se describe la etapa de procesos de software, donde se describen los modelos de desarrollo y el diseño centrado en el usuario. Luego, se aborda el tema de la ingeniería de requerimientos. Seguido, se describe la etapa del diseño (decisiones y estilos arquitectónicos, diagramas de flujo de datos) y codificación del software

(características del código y reutilización de software). Finalmente, se describe la etapa de pruebas de software, donde se representan las técnicas de pruebas.

- ✓ **El capítulo VI**, corresponde a la quinta parte de la revisión bibliográfica, donde se presenta y describe una selección de metodologías utilizadas para el diseño de juegos serios y los criterios de evaluación para poder analizarlas. El capítulo se compone concretamente por la introducción, definición de criterios de análisis, recopilación y descripción de cada una las metodologías seleccionadas y, el análisis de forma individual según los criterios definidos. Posteriormente, se aborda el análisis y comparación de las metodologías recopiladas acorde a los criterios de evaluación previamente definidos, resultados en función del trabajo realizado tanto en la revisión teórica como en el análisis.
- ✓ **El capítulo VII**, corresponde a la sexta y última parte de la revisión bibliográfica, en la cual se aborda y describe cada una de las arquitecturas seleccionadas que han sido utilizadas para el diseño de juegos serios. Luego, se presenta el análisis de resultados obtenidos a partir de la aplicación de los criterios de evaluación.
- ✓ **El capítulo VIII**, presenta y describe una propuesta metodológica y arquitectónica para el diseño de prototipos de juegos serios orientados a la adquisición de habilidades o competencias tecnológicas. Específicamente, el capítulo está conformado por la introducción y la descripción del procedimiento de implementación de la propuesta (pasos metodológicos y arquitectónicos para el desarrollo de juegos serios). Luego, se describe en detalle cada una de las 7 fases que componen la metodología INTEGRA+506.
- ✓ **El capítulo IX**, presenta y describe en detalle la aplicación de los pasos metodológicos propuestos en INTEGRA+506 para el diseño del prototipo no funcional del juego serio denominado SkillNet, el cual está orientado a la

formación de competencias tecnológicas en el profesorado. Finalmente, se describe el proceso realizado para la validación de la metodología.

- ✓ El capítulo X, presenta las conclusiones y, se aborda brevemente las limitaciones durante el proceso. Luego, se describe las líneas de trabajos futuros del estudio con base en los hallazgos de investigación y el grado de cumplimiento de los objetivos propuestos. Finalmente, se enlista la producción científica relacionada y derivada de los resultados parciales de la tesis y del proceso de formación durante el doctorado.

# **CAPÍTULO II. TIC y COMPETENCIAS TECNOLÓGICAS**

## 2.1. Introducción

En este capítulo se presenta la conceptualización de las TIC y competencias tecnológicas, se realiza un abordaje desde el punto de vista de varios autores. Luego, se describen sus principales características. Finalmente, se realiza un breve recorrido y descripción del impacto del desarrollo de competencias o habilidades tecnológicas en educación.

Los contenidos propuestos a desarrollar en este capítulo permiten brindar respuesta a las preguntas de investigación; Plx1 y Plx2 establecidas para el estudio y enumeradas en el Capítulo 1:

- Plx1 ¿Cómo se define o describe el concepto de Tecnologías de la Información y la Comunicación (TIC)?
- Plx2 ¿Cómo se define el concepto de competencias o competencias tecnológicas?

Es importante indicar que para efectos de esta investigación, se retoma de forma breve estas definiciones (TIC, competencias y competencias tecnológicas), ya que las mismas han sido descritas y analizadas con mayor detalle en los trabajos de investigación previos realizados por el tesis, mismos relacionados con la temática en estudio (Sandí-Delgado et al., 2018; Sandí-Delgado & Sanz, 2018a).

## 2.2. Conceptualización TIC, competencias y competencias tecnológicas

En esta sección, se relevan las definiciones de las TIC, competencias y competencias tecnológicas, mismas desde el punto de vista de diferentes autores.

### 2.3.1. Definición y caracterización de las TIC

La conceptualización y caracterización de las Tecnologías de la Información y la Comunicación (TIC) ha variado en el transcurso del tiempo, dependiendo del autor que las analiza y define, por ende, las definiciones acotadas pueden variar según la especialización y área de formación profesional que el autor posea en la temática.

En la [Tabla 2.1](#), se realiza un relevamiento de las principales definiciones acotadas por diferentes autores expertos en el área.

*Tabla 2.1. Definiciones del Término TIC*

Autor/es	Definición
Cruz-Pérez et al. (2019, p. 5)	Las TIC son la creación educativa de esta era y permiten tanto a docentes como a estudiantes cambios determinantes en el quehacer diario del aula y en su proceso de enseñanza-aprendizaje.
Tapia (2018, p. 5)	Entenderemos como TIC a todo producto de la técnica que favorezca la creación, resguardo o consumo de recursos electrónicos o digitales.
González (2017, p. 5)	TIC refiere al hardware y software que brinda la posibilidad de interactuar con datos (editar, crear, almacenar o transmitir), mediante distintos sistemas de información que integran medios de informática, telecomunicaciones o redes. Además, permite la comunicación interpersonal o multidireccional y posibilita un impacto sustantivo al conocimiento.
Cruz et al. (Cruz-Alvarado et al., 2017)	Se podría definir las TIC como todas aquellas herramientas y recursos tecnológicos que permiten gestionar de una forma diferente a la tradicional la información, en las cuales se posibilita la oportunidad de crear, procesar, almacenar, representar, utilizar e integrar la información de una forma interactiva.
Hernández-Doria et al. (2014, p. 4)	Como tecnologías de la información y comunicación se pueden mencionar todos aquellos recursos tecnológicos utilizados para gestionar, procesar, almacenar y presentar información, los cuales se han convertido en instrumentos que facilitan muchas de las labores que desarrollan los seres humanos en la actualidad, viéndose esto reflejado en la función que cumplen en la mediación de los procesos de enseñanza-aprendizaje.
Cobo (2009, p. 312)	Tecnologías de la Información y la Comunicación (TIC): Dispositivos tecnológicos (hardware y software) que permiten editar, producir, almacenar, intercambiar y transmitir datos entre diferentes sistemas de información que cuentan con protocolos comunes. Estas aplicaciones, que integran medios de informática, telecomunicaciones y redes, posibilitan tanto la comunicación y colaboración interpersonal (persona a persona) como la multidireccional (uno a muchos o muchos a muchos). Estas herramientas desempeñan un papel sustantivo en la generación, intercambio, difusión, gestión y acceso al conocimiento.
Roblizo & Cózar (2015, p. 23)	Fenómeno revolucionario, impactante y cambiante, que abarca tanto lo técnico como lo social y que impregna todas las actividades humanas, laborales, formativas, académicas, de ocio y consumo.

Tal como se puede apreciar en la [Tabla 2.1](#), el concepto TIC ha sido definido ampliamente por los diferentes autores citados, los cuales han resaltado diferentes características en común de las TIC, tales como: **recursos tecnológicos** (referente al hardware; dispositivos físicos y, al software; aplicaciones y herramientas), **gestión**

**del conocimiento** (tratamiento de los datos e información; almacenamiento, procesamiento, intercambio y trasmisión de la información), y otras características que pueden ayudar a comprender más ampliamente el concepto, tal como la colaboración, la innovación e interacción.

Así, para efectos de esta investigación se brinda una definición propia de las TIC; las cuales podrían entenderse como:

Un conjunto de **recursos tecnológicos** (hardware y software) que posibilitan la **gestión del conocimiento** (tratamiento de la información) de forma **innovadora, interactiva y colaborativa**, el cual podría potenciar el desarrollo de **procesos formativos mediatizados**.

Finalmente, se debe tener presente que las TIC por sí mismas, no pueden cambiar los sistemas y metodologías educativas, sino que, estos cambios van a depender del correcto uso e integración que se les dé a las tecnologías digitales (Boulahrouz et al., 2019).

En la próxima subsección se describe en detalle la definición del concepto de competencias acorde a diferentes autores que las definen.

### 2.3.2. Definición de competencias

Según Bezanilla et al. (2014) las competencias son un conjunto integrado de habilidades, donde los conocimientos, las actitudes, las normas y los valores forman un componente importante. Igualmente, Cárdenas-García et al. (2016) señalan que el concepto de competencias abarca una variedad de comportamientos requeridos para el ejercicio de una ocupación, lo cual va más allá de sólo poseer las capacidades propias de una actividad profesional. Al respecto Ochoa et al. (2016, p. 238.e2) plantean que el propósito principal de las competencias radica en “*lograr un aprendizaje que integre el saber, el saber hacer, el saber ser y el saber estar*”.

Para ampliar el concepto y facilitar su comprensión, la Tabla 2.2 enlista una serie de definiciones de competencias según autores de renombre que las definen.

Tabla 2.2. Definiciones del término competencias

Autor/es	Definición
Ferri (2013, p. 37)	La capacidad comprobada para utilizar los conocimientos, habilidades y capacidades personales, sociales o metodológicas, en situaciones de trabajo o estudio y en el desarrollo profesional y personal.

<b>Autor/es</b>	<b>Definición</b>
Muñoz-Osuna et al. (2016, p. 126)	El término competencias refiere a una integración de conocimientos, actitudes y habilidades que permiten desempeñarse exitosamente en un conjunto de funciones.
Sandí & Sanz (Sandí-Delgado & Sanz, 2018b)	El término de competencias está relacionado con cualidades humanas que integran funcionalmente el saber (diferentes tipos de conocimientos), saber hacer (hábitos, destrezas capacidades y habilidades) y, saber ser (valores y actitudes) que permite a las personas desempeñarse de una forma idónea en diferentes entornos, sean estos personales, educativos o laborales.
Baca (2015, p. 237)	Competencia se define como la capacidad del ser humano para realizar un conjunto de acciones, mediante la articulación de sus múltiples recursos personales (actitudes, conocimientos, emociones, habilidades, valores...) con el propósito de lograr una respuesta satisfactoria a un problema planteado en un contexto determinado.
Cataldi et al. (2010, p. 217)	Las competencias se definen como las habilidades, conocimientos, actitudes, capacidades, valores, comportamientos y en general atributos personales, que se relacionan más directamente con el éxito de las personas en sus trabajos, funciones y relaciones.
Padilla-Zea et al. (2015, p. 26)	Entendemos por competencias los conocimientos, actitudes, habilidades, y destrezas que desarrolla una persona para comprender, transformar y llevar a cabo acciones en el mundo en el que se desenvuelve.
Tejeda & Sánchez del Toro (2010, p. 41)	La competencia es una cualidad humana que se configura como síntesis dialéctica en la integración funcional del saber (conocimientos diversos), saber hacer (habilidades, hábitos, destrezas y capacidades) y saber ser (valores y actitudes) y que son movilizados en un desempeño idóneo a partir de los recursos personológicos del sujeto, que le permiten saber estar en un ambiente socioprofesional y humano en correspondencia con las características y exigencias complejas del entorno.

En la próxima subsección se aborda en detalle la definición del término de competencias tecnológicas, asimismo, se identifican aspectos o características comunes entre las definiciones acotadas, luego, se brinda una definición propia al término de competencias tecnológicas.

### 2.3.3. Definición de competencias tecnológicas

El término competencias tecnológicas<sup>8</sup> también es reconocido como competencias TIC o competencias digitales (Baca, 2015; Cruz-Rodríguez, 2019; Lázaro-Cantabrina et al., 2018; Prendes et al., 2018; Rodríguez-García et al., 2019), el cual ha sido ampliamente definido por diferentes autores (Arevalos, 2014; Cataldi et al., 2010; Choi et al., 2018; Cuberos de Quintero & Vivas, 2017;

<sup>8</sup> Para efectos de esta investigación, el término “tecnológicas” refiere a tecnologías digitales. Es importante realizar la aclaración debido a que el concepto de tecnología es más amplio y no se circunscribe únicamente a tecnologías digitales, sino que puede abarcar tecnologías organizativas, tecnologías simbólicas y artefactuales (Sancho, 1998; Sandí-Delgado & Sanz, 2018b)

Nousiainen et al., 2018; Prendes et al., 2010; Prendes & Gutiérrez, 2013; Sandí-Delgado, 2019; Silva et al., 2016; Tondeur et al., 2018), quienes lo han asociado al uso y dominio de variedad de herramientas tecnológicas, además, a las habilidades asociadas y requeridas para su correcto uso e integración de las mismas en las diferentes áreas del conocimiento.

Autores como Lázaro-Cantabrana et al. (2018), Silva et al. (2016) y Ferrari (2013), señalan que las competencias tecnológicas son consideradas como competencias claves, destacadas y transversales, debido a que potencian la formación o adquisición de otras competencias, tales como la lingüística y la matemática, requeridas por la ciudadanía para garantizar una participación activa en la sociedad y la economía.

Al respecto, el Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF, 2017), señala que las competencias tecnológicas están relacionadas con el uso creativo, seguro y crítico de las TIC para alcanzar los objetivos afines al entretenimiento, la comunicación, el aprendizaje, al trabajo, la inserción y contribución en la sociedad, apoyadas en las competencias básicas relacionadas al uso de tecnologías digitales.

En esta misma línea, se identificó que en determinados países de Iberoamérica se han definido perfiles de competencias tecnológicas esperadas en el profesorado, mismos clasificados en dimensiones e indicadores, en el caso de **Chile** es el Ministerio de Educación de Chile (2006), en el caso de España, investigadores como Prendes & Gutiérrez (2013) que retoman esas definiciones del Ministerio de Educación de Chile y las refinan, en el caso de **Colombia** (Campo et al., 2013; Hernández et al., 2016), **Uruguay** (Silva et al., 2016), **Paraguay** (Arevalos, 2014) y **Costa Rica** (UCR, 2004) son presentadas fundamentalmente por investigadores. Además, se encontró que la definición de competencias tecnológicas tiene que ver con el **saber, saber hacer, saber estar y saber ser**.

No obstante, el uso de esta terminología puede variar de acuerdo con diferentes autores que las analizan (según su área de formación, especialización o propósito de la investigación). Con la finalidad de ampliar el concepto, en la [Tabla 2.3](#), se enumeran algunas de definiciones brindadas por diferentes autores al concepto de competencias tecnológicas.

Posterior al análisis realizado de las definiciones de competencias presentadas con anterioridad en la Tabla 2.2 y Tabla 2.3, se identificaron aspectos o características comunes entre las definiciones acotadas, entre las cuales se destacan:

- ✓ **Conocimiento** - Se refiere a los saberes propios con los cuales cuenta la persona para ejercer un oficio o profesión.
- ✓ **Actitudes/capacidades** - Se refiere al comportamiento de un individuo ante alguna circunstancia y la forma de abordar la resolución de esta.
- ✓ **Valores** - Relacionados con el saber ser; con el comportamiento de las personas, los cuales posibilitan la toma de decisiones con el propósito de ser mejor a nivel personal. Se relacionan con las actitudes.
- ✓ **Habilidades/destrezas** - Se refiere a la facilidad para adquirir conocimientos nuevos o bien, desarrollarse en una actividad. En este caso, adquirir conocimientos en tecnologías. Asociado al saber hacer, es decir, realizar alguna actividad de la mejor forma (Sandí-Delgado et al., 2018).

**Tabla 2.3. Definiciones del término competencias tecnológicas**

Autor/es	Definición
Rodríguez-García et al. (2019, p. 66)	Entendemos por competencia digital la capacitación de saber utilizar la tecnología de manera eficaz para mejorar todas las áreas de nuestra vida diaria. La competencia digital no se trata de una habilidad aislada a desarrollar, sino que toda ella supone un compendio de destrezas, habilidades y actitudes ante diferentes áreas y dimensiones de conocimiento.
Unión Europea (2006, p. 394/15)	La competencia digital entraña el uso seguro y crítico de las tecnologías de la sociedad de la información (TSI) para el trabajo, el ocio y la comunicación. Se sustenta en las competencias básicas en materia de TIC: el uso de ordenadores para obtener, evaluar, almacenar, producir, presentar e intercambiar información, y comunicarse y participar en redes de colaboración a través de Internet.
Bozu & Canto (2009, p. 91)	El conjunto de conocimientos, habilidades, actitudes y valores necesarios para realizar una docencia de calidad. Esto es, lo que han de saber y saber hacer los profesores/as para abordar de forma satisfactoria los problemas que la enseñanza les plantea.
Cuberos de Quintero & Vivas (2017, p. 3)	El conjunto de saberes (conceptos, habilidades y actitudes...) que deben ser adquiridos, trabajados y aplicados para el desenvolvimiento eficaz en el proceso de formación y comunicación mediado por las TIC.

Autor/es	Definición
Prendes & Gutiérrez (2013, p. 199)	La competencia tecnológica -competencia digital o competencia TIC- tiene que ver con la capacidad, el conocimiento y la actitud sobre el uso de tecnologías de la información y la comunicación en sus diversas funciones y contextos de aplicación.
Campos, Brenes & Solano (2010, pp. 8–9)	Estas competencias son referidas al dominio de herramientas tecnológicas y a las habilidades asociadas con el uso de la tecnología, incluyen la posibilidad de seleccionar e incorporar los cambios y los nuevos desarrollos tecnológicos para potenciar procesos educativos mediante la interacción con el estudiantado y con el cuerpo docente.

A partir de las características comunes identificadas en las definiciones analizadas y, tomando en cuenta que en los diferentes perfiles de competencias tecnológicas esperadas del profesorado definidos en varios países de Iberoamérica tienen que ver con el Saber (conocimientos), Saber hacer (habilidades), Saber estar y Saber ser (actitudes y valores), se adopta una definición propia del término competencias tecnológicas, en la cual, se toma como base de referencia la definición propia acotada en una investigación previa realizada por Sandí, Sanz & Lovos (2018), misma que se retoma y amplia para esta investigación. Entonces, las competencias tecnológicas podrían ser definidas como:

Un conjunto de **saberes** relacionados con el correcto **uso, dominio e integración** de las **tecnologías digitales** y otras áreas del conocimiento relacionadas, que a través de diferentes **habilidades, actitudes y valores** permiten a las personas el **saber ser**, es decir, desarrollar un comportamiento propio ante un evento para ser capaz de potenciar el **mejoramiento personal, académico o profesional**, y con ello, garantizar su participación en la sociedad.

En la próxima sección, se describe la importancia del desarrollo o formación de competencias tecnológicas en el ámbito educativo.

#### 2.4. Importancia del desarrollo de competencias o habilidades tecnológicas en educación

El profesorado cumple un rol importante que conlleva la responsabilidad y la sensibilidad para el correcto uso e integración de las TIC en los procesos formativos (Farjon et al., 2019). Lo anterior, debido que en la actualidad el profesorado utiliza las TIC como un medio para apoyar el proceso de enseñanza y aprendizaje (Fraillon et al., 2014), por tal razón, el profesorado debe ser tecnológica y pedagógicamente competente, lo cual coincide con lo explicitado en numerosas investigaciones

(Blackwell et al., 2016; Dalal et al., 2017; Holland & Piper, 2016; Nelson et al., 2019; Rosenberg & Koehler, 2015; Sampiero & Barragán, 2018; Scherer et al., 2017, 2018; Tseng et al., 2019; Yeh et al., 2017), en las cuales se considera que para que se dé una correcta integración de las TIC por parte del profesorado, el mismo debe poseer habilidades tecnológicas, pedagógicas y disciplinares.

Habilidades y conocimientos que han sido planteadas por el Modelo de Conocimiento Técnico Pedagógico del Contenido (*Technological pedagogical content knowledge*, TPACK), el cual ha sido definido como un modelo de extensión del conocimiento del contenido pedagógico al cual se le ha añadido el contenido tecnológico (Almerich et al., 2016; Sandí-Delgado & Sanz, 2018b).

Tal como se ha mencionado, la formación en habilidades o competencias TIC ha resultado ser esencial en diferentes áreas del conocimiento y en particular, en el ámbito educativo. De acuerdo con García-Fallas (2004) y Ruiz-Requies et al. (2010) las TIC pueden ser utilizadas para favorecer la generación de ambientes de aprendizaje mediados con recursos tecnológicos.

En este sentido, las IES enfrentan el desafío de transformar los ambientes de enseñanza y las metodologías de aprendizaje que permitan la incorporación amplia y dinámica de las TIC en los procesos formativos (Pérez & Salas, 2009; Sandí-Delgado & Cruz-Alvarado, 2016). Además, garantizar los mecanismos y recursos necesarios (económicos, tecnológicos, infraestructura, humanos) para que el profesorado pueda participar en procesos de actualización o formación que les permita potenciar sus habilidades tanto a nivel pedagógico como tecnológico necesarias y requeridas para guiar, orientar y facilitar el aprendizaje del alumnado (Morales, 2013; Yu & Prince, 2016; Zempoalteca et al., 2017).

## 2.5. Conclusiones del capítulo

El capítulo presentó la conceptualización a partir de diferentes autores de las Tecnologías de la Información y la Comunicación (TIC), competencias y competencias tecnológicas, además, se ofreció nuestra propia definición de cada concepto a partir de los aspectos comunes y claves identificados en las terminologías analizadas, mismas, que sirven de referente para comprender con mayor amplitud los conceptos desarrollados. El capítulo ostenta las bases conceptuales para el desarrollo posterior de la tesis.

Seguido, se realiza un abordaje en relación con la importancia del desarrollo de las competencias tecnológicas por parte del profesorado en educación superior y, cómo dichas competencias podrían ayudar en la mejora de los procesos de enseñanza y aprendizaje, el cual es un tema de relevancia en esta investigación.

# CAPÍTULO III. JUEGOS SERIOS

### 3.1. Introducción

Este capítulo presenta la conceptualización de juegos serios tomando en cuenta los aportes de varios autores, luego se adopta una definición o conceptualización propia. Seguido, se realiza una caracterización a nivel general de los juegos serios y se abordan algunas características funcionales de los mismos, se realiza un recorrido desde el punto de vista de varios autores. Por último, se abordan antecedentes de uso de juegos serios en procesos de desarrollo de competencias tecnológicas.

Los contenidos planteados a desarrollar en este capítulo permiten brindar respuesta a las preguntas de investigación; Plx3, Plx4, Plx5 y Plx6 establecidas para el estudio y enumeradas en el Capítulo I:

Plx3 ¿Cómo se definen actualmente los juegos serios?

Plx4 ¿Cuáles son las principales características, requerimientos técnicos o funcionales de los juegos serios?

Plx5 ¿Qué antecedentes existen de la utilización de juegos serios para la generación y desarrollo de competencias tecnológicas?

Plx6 ¿Cuáles metodologías se han utilizado para el desarrollo de juegos serios?, ¿qué antecedentes existen en este sentido?

Al igual que en el Capítulo II, es conveniente indicar que para efectos de esta investigación, el abordaje que se realiza a la definición del término juegos serios, es breve, ya que dicho término ha sido definido y caracterizado con amplitud en las investigaciones previas realizadas por el tesista (Sandí-Delgado et al., 2018; Sandí-Delgado & Bazán, 2017; Sandí-Delgado & Sanz, 2018a).

### 3.2. Conceptualización de juegos serios

Fue hasta en 1970 que se escuchó por primera vez mencionar el concepto de juegos serios (SG, *serious games*), el cual fue introducido por Abt (1970) a través de la publicación de su libro titulado “*Serious Games*”, impreso por la editorial *Viking Press*. Posteriormente, tanto el término como el autor han sido referenciados en diferentes investigaciones (Archuby et al., 2017b; Marcano, 2008; Michael & Chen,

2006; Sandí-Delgado et al., 2018; Sandí-Delgado & Bazán, 2017; Sandí-Delgado & Sanz, 2018b; Vogel, 1970), entre otros.

La intención principal del libro “*Serious Games*”, se fundamentó en dar a conocer al público en general las posibilidades que ofrecían los juegos serios para la formación o adquisición de nuevas habilidades en las diferentes áreas del conocimiento, ya fuese a través del juego, la instrucción, la simulación, la formación o la educación y, primordialmente, sus potencialidades para proporcionar placer y motivación (Sandí-Delgado et al., 2018; Sandí-Delgado & Sanz, 2018a).

Con el propósito de ampliar el concepto y facilitar su comprensión, se brinda a continuación una serie de definiciones del término de juegos serios.

**Según** (Massa & Kühn, 2018b, p. 1)

Los Serious Games (SG) son aplicaciones interactivas creadas con una intencionalidad educativa, que proponen la explotación de la jugabilidad como experiencia del jugador. Presentan a los jugadores retos y misiones que implican tomas de decisiones, resolución de problemas, búsqueda de información selectiva, cálculos, exploración, experimentación, competencia, cooperación etc., logrando el efecto inmersivo en el juego, como una prolongación de la experiencia vital del usuario.

**Según** (Marsh, 2011, p. 61)

Se utiliza una creciente variedad de categorías para encapsular los propósitos emergentes de los juegos serios, como el aprendizaje, la capacitación, la educación, la salud, para el cambio, la persuasión, o simplemente por experiencia o emoción.

**Según** (Michael & Chen, 2006, p. 21)

Aquellos juegos que esencialmente poseen un propósito educativo explícito y cuidadosamente pensado. Su propósito principal no es el entretenimiento, el disfrute o la diversión. Eso no quiere decir que los juegos bajo el paraguas de los juegos serios no sean entretenidos, agradables o divertidos. Sino que tienen otro propósito, ofrecer un aprendizaje.

**Según** (Girard et al., 2013, p. 207)

Los juegos serios han surgido con la intención de apoyar diferentes áreas del conocimiento, en la cual se realiza una combinación del componente lúdico con diferentes fines, entre ellos, el aprendizaje.

**Según** (Marcano, 2008, p. 98)

Se le asigna este nombre a un grupo de videojuegos y simuladores cuyo objetivo principal es la formación antes que el entretenimiento. Esta área de desarrollo y creación de videojuegos ha surgido como una manera inteligente de combinar los beneficios de los videojuegos, su poder de penetración en la población y las necesidades de educación y formación efectiva tanto a nivel político-institucional como empresarial y comercial.

**Según** (Calabor et al., 2017, pp. 2–3)

El término serious games (SG) hace referencia a los juegos utilizados en la formación y educación, sean estos simulaciones, videojuegos, mundos virtuales o realidad aumentada. Muchos autores consideran las TIC en general y los SG en particular, como métodos útiles para apoyar el proceso de enseñanza-aprendizaje y como recursos adecuados para el desarrollo de competencias. El juego serio debe centrar la atención del jugador en dos elementos fundamentales: aprendizaje y entretenimiento.

**Según** (Archuby et al., 2017b, p. 14)

Son aquellos que poseen un propósito educacional explícito y cuidadosamente pensado, y no han sido concebidos para ser jugados principalmente como modo de entretenimiento.

**Según** (Díaz et al., 2015, p. 1)

Juegos que tienen una finalidad educativa, de entrenamiento o de información y están cuidadosamente pensados para tal fin.

Acorde a las definiciones anteriores y a lo expuesto por diferentes investigadores (Michael & Chen, 2006; Muñoz et al., 2011; Sandí-Delgado et al., 2018; Sandí-Delgado & Bazán, 2017; Sandí-Delgado & Sanz, 2018a; Yusoff et al., 2010) los juegos serios han sido utilizados para la formación y el cambio de comportamiento en diferentes áreas del conocimiento; salud, gobierno, ONG's, la

economía, entrenamiento militar, la educación, la generación de habilidades y competencias tecnológicas.

En esta misma línea, otros autores (Boyle et al., 2012; Connolly et al., 2012; Kwon & Lee, 2016; Massa & Kühn, 2018a; Sawyer & Smith, 2008) agregan que los juegos serios pueden ser utilizados para potenciar procesos formativos, debido a sus características vinculadas al campo educativo referentes a la motivación, la interactividad, la participación activa del estudiantado y profesorado, donde su componente lúdico y pedagógico podrían permitir un aprendizaje a partir del error (*feedback*).

A partir de las definiciones brindadas al inicio de esta sección (3.2), las características abordadas en la sección 3.3 y, de tomar como referencia o base la definición propia de juegos serios ofrecida previamente por Sandí & Sanz (2018a) la cual es retomada y ampliada, luego, se procede a redefinir el concepto de juegos serios, así, la definición adoptada para esta investigación consiste en que:

Un juego serio se podría definir como aquel juego que tiene **un enfoque que va más allá del entretenimiento o la diversión**, el cual posee un **propósito educativo explícito** y cuidadosamente pensado, es decir, se puede utilizar para **potenciar diferentes áreas del conocimiento**, ofrecer un aprendizaje, apoyar procesos formativos, promover cambios de actitud o comportamiento, **generar emociones, potenciar la adquisición de habilidades o competencias**, entre otros. **Caracterizado principalmente** por ser atractivo, interactivo, entretenido-dinámico, motivador, desafiante, fácil de utilizar, con la capacidad de estimular la colaboración, el razonamiento y el pensamiento crítico.

Con esta definición de juegos serios, es importante atender a esta posibilidad de definir una metodología y arquitectura de software para la producción de juegos serios orientados a promover la adquisición de conocimientos y habilidades relacionadas con las competencias tecnológicas.

### **3.3. Características de juegos serios**

Las características de los juegos serios que se describen a continuación han sido identificadas a partir de los hallazgos de las investigaciones de revisión bibliográfica realizadas por (Sandí-Delgado et al., 2018; Sandí-Delgado & Bazán,

2017; Sandí-Delgado & Sanz, 2018a) en las cuales se analizó los resultados de otras investigaciones publicadas por investigadores expertos en el área (Chipia, 2011; Girard et al., 2013; Bezanilla et al., 2014; Del-Moral & Guzmán-Duque, 2014; Guenaga, Eguíluz, Rayón, & Quevedo, 2015; Hainey, Connolly, Boyle, Wilson, & Razak, 2016; Tsai, Huang, Hou, Hsu, & Chiou, 2016; Angelini, García-Carbonell, & Martínez-Alzamora, 2017), quienes han identificado, discutido y analizado una serie de atributos que permiten la caracterización de los juegos serios.

A continuación, se resumen las características identificadas de los juegos serios en las investigaciones de (Sandí-Delgado, 2019; Sandí-Delgado et al., 2018; Sandí-Delgado & Bazán, 2017; Sandí-Delgado & Sanz, 2018a).

- ✓ **Motivación intrínseca** - Provocan sensaciones y experiencias, que buscan que el jugador siga jugando, lo cual facilita alcanzar los objetivos no lúdicos.
- ✓ **Objetivos formativos** - Presentan objetivos o intereses educativos claros en sus contenidos. Son divertidos debido a la interactividad y motivación que suelen proponer.
- ✓ **Potencian competencias y habilidades** - Incrementan el desarrollo de nuevas competencias y habilidades como la colaboración, la solución de problemas, la reflexión, la comunicación, el pensamiento crítico o la alfabetización digital.
- ✓ **Aplicación de conocimientos** - Permiten al usuario (jugador) no sólo aprender, sino también demostrar y aplicar lo que ha aprendido.
- ✓ **Emotividad** - Impactan emocionalmente a los jugadores, éstos mejoran su autoestima.
- ✓ **Colaboración** - Animan a la colaboración y, en cierto nivel, son similares a los entornos de aprendizaje colaborativos.
- ✓ **Interactividad** - Potencian el desarrollo de ambientes interactivos donde el jugador puede tener el control de su propio aprendizaje a la vez que experimenta en distintos escenarios.
- ✓ **Psicomotricidad** - Fomentan el control psicomotriz, que permite al jugador coordinar lo que piensan con lo que están haciendo.
- ✓ **Dinamización** - Los juegos serios educativos permiten dinamizar la conducta y el pensamiento, que hace que los jugadores tengan mayor capacidad de respuesta y agilidad mental.

- ✓ **Imaginación y pensamiento** - Impulsan la imaginación y el pensamiento, ya que los jugadores están en un mundo imaginario, donde se desarrolla la acción.
- ✓ **Simulación** - Permiten simular y ser utilizados para realizar experimentos peligrosos en la vida real, como aquellos que pueden poner en riesgo la seguridad e integridad física y emocional de las personas.

Aunado a lo anterior, una investigación realizada por (Sandí-Delgado & Bazán, 2017, 2019), la cual tenía como propósito valorar las posibilidades de ofrecer los juegos serios como un software de servicio, se identificaron como parte de los resultados una serie de características funcionales que los juegos serios adquirirían al ser utilizados como software de servicio en educación superior, mismas que se resumen en el siguiente *check list*:

- ✓ Ser una aplicación de tipo web con acceso desde cualquier browser con conexión a internet y altamente disponible.
- ✓ Ser de arquitectura *multi-tenant* (separación lógica de datos de cada usuario).
- ✓ Ser una aplicación altamente colaborativa entre usuarios.
- ✓ Ser una aplicación altamente disponible desde cualquier lugar y momento.
- ✓ Ser de versión simple y compartida para todos los clientes.
- ✓ Ser personalizables y configurables a nivel de cada perfil de usuario.
- ✓ Ser escalables y con mecanismos de balanceo de carga.
- ✓ Ofrecer interactividad totalmente sincrónica.
- ✓ Ofrecer disponibilidad y soporte multiplataforma.
- ✓ Ofrecer seguridad y control de acceso por usuario.
- ✓ Ofrecer integración de aplicaciones (manteniéndose de forma independiente entre sí).
- ✓ Ofrecer soporte para el aprovisionamiento de recursos para los suscriptores nuevos.
- ✓ Ofrecer soporte para la suscripción y facturación (monetizaciones, facturaciones, monitoreos, manejo de cuentas de usuario, loggings, control de uso y métricas).

- ✓ Ofrecer ubicuidad, servicios y recursos multimediales en tiempo real.
- ✓ Ofrecer guía o ayuda en diferentes idiomas.

Según Sandí & Bazán (2017), Sandí, Sanz & Lovos (2018) y Sandí-Delgado & Bazán (2021) las características de los juegos serios identificadas, tienden a ser utilizados en la formación y entrenamiento en habilidades y competencias determinadas, las cuales pueden contribuir a la generación de emociones y sensaciones, sirven de apoyo para comprender diferentes procesos políticos, sociales, religiosos, de salud, económicos y educativos. Por ejemplo, la vida diaria está apoyada por emociones y, en el aprendizaje esas emociones se marcan aún más; por ende, los juegos serios podrían ser una alternativa para la formación y adquisición de habilidades o competencias en diferentes áreas del conocimiento, ya que, a como lo describen las características citadas, los juegos serios pueden generar emotividad, motivación intrínseca, interactividad, características importantes que podrían potenciar la imaginación y el pensamiento crítico, necesarios para la generación de nuevos conocimientos.

Finalmente, se podría indicar que los juegos serios han sido ampliamente analizados y caracterizados, estas características podrían variar y quedan sujetas acorde al interés, propósito y objetivos de cada investigador que las estudia y analiza.

### **3.4. Antecedentes de uso de juegos serios en procesos de desarrollo de competencias TIC**

En las investigaciones previas realizadas por el tesista (Sandí-Delgado et al., 2018; Sandí-Delgado & Bazán, 2017; Sandí-Delgado & Cruz-Alvarado, 2018) se han identificado experiencias de uso de diferentes juegos serios para potenciar la formación y adquisición de competencias o habilidades tecnológicas en los jugadores en diferentes países, que se describen brevemente a continuación.

En **Argentina**, se ha evidenciado la utilización de dos juegos que han sido diseñados y desarrollados en la Facultad de Informática de la Universidad Nacional de La Plata (UNLP). El primero, denominado *RITA* (*Robot Inventor to Teach Algorithms*). Este juego serio fue diseñado por Vanessa del Carmen Aybar Rosales como parte de la investigación para alcanzar el grado de Licenciatura en

Informática, la investigación estuvo dirigida por la Lic. Claudia Queiruga y Lic. Claudia Banchoff. (Sandí-Delgado et al., 2018; Sandí-Delgado & Sanz, 2018a).

El juego tiene como objetivo potenciar la formación de habilidades y conocimientos tecnológicos, específicamente, en el área de la programación por bloques, mediante la estimulación de la lógica de la programación y la algoritmia; permite al jugador adquirir conocimientos básicos en programación orientada a objetos y conocimientos del lenguaje Java (Aybar et al., 2012; Queiruga et al., 2012).

El segundo juego serio diseñado en **Argentina** por la UNLP es AstroCódigo, el cual fue diseñado y desarrollado por Javier Bione y Pablo Miceli como proyecto final de tesis de grado. La investigación estuvo dirigida por la Dra. Cecilia Verónica Sanz y la Lic. Verónica Artola. El objetivo principal del juego consiste en acercar al jugador a los conceptos básicos de programación; secuencias de instrucciones, estructuras de control y, algoritmia, en general (Sandí-Delgado et al., 2018; Sandí-Delgado & Sanz, 2018a). AstroCódigo potencia la formación de competencias tecnológicas, en programación y resolución de problemas a través de la utilización de secuencias de instrucción y estructuras de control, las cuales permiten el arrastre de bloques para la programación de algoritmos (Bione et al., 2017b, 2017a; Sandí-Delgado & Sanz, 2018a; Sanz, Artola, et al., 2018).

Un tercer juego en **Argentina**, igualmente desarrollado en la UNLP con fines pedagógicos es FraccionAR, el cual fue diseñado en el Instituto de Investigación en Informática III-LIDI por el equipo de trabajo conformado por la Dra. Cecilia Verónica Sanz, Ing. Mauricio Nordio y la Lic. Verónica Artola. *FraccionAR* es un juego serio basado en interacción tangible, desarrollado para el aprendizaje y comprensión de fracciones por parte de los estudiantes a nivel de primaria. El juego busca potenciar competencias pedagógicas relacionadas con el concepto de fracciones de una manera lúdica y a través de la experiencia (Sanz, Nordio, et al., 2018).

Igualmente, se registra un cuarto juego serio diseñado en **Argentina** por la UNLP a nivel pedagógico, el cual se denomina Desafíate. Este juego serio fue desarrollado por Federico Archuby como parte de la investigación de su tesis para alcanzar el grado de Licenciatura en Sistemas, la investigación estuvo dirigida por la Dra. Cecilia Sanz y la Dra. Mabel Pesado. El juego está orientado para dispositivos móviles, el cual se integra a las posibilidades del entorno virtual de enseñanza y

aprendizaje IDEAS, a partir de las herramientas de autoevaluación que este EVEA dispone. *Desafíate* permite potenciar el desarrollo de competencias pedagógicas, específicamente relacionadas con acompañar a la autoevaluación del estudiantado en la aplicación de conceptos y recuperación del conocimientos abordados en determinados procesos educativos (Archuby et al., 2017a, 2017c, 2018).

En **Irlanda**, se ha diseñado y desarrollado el juego serio GSD Sim, el cual es un juego serio educativo del tipo simulación, desarrollado por los investigadores Tom Mason, Kevin Farrell, Miles McGuire y Ross McKinley del “Centro de Investigación de Ingeniería de Software de Irlanda”.

El objetivo principal de GSD SIM consiste en potenciar las habilidades de los jugadores en el desarrollo de software global (GSD, *Global Software Development*). El juego posibilita al jugador vivenciar de una forma más realista, divertida y entretenida las dificultades que implica el GSD desde el punto de vista de un gerente de proyecto, en un tiempo mucho más corto y a un costo menor que un proyecto de GSD real, permitiendo potenciar la mejora en habilidades y destrezas, formación de competencias tecnológicas, trabajo colaborativo y el *feedback* (Clear et al., 2016; Noll et al., 2010, 2014; Sandí-Delgado & Sanz, 2018a).

En **Francia**, se creó el juego serio Tamagocours, desarrollado por Erick Sánchez del Instituto de Educación Francés. *Tamagocours* es un juego colaborativo en línea desarrollado con el propósito de ser utilizado para entrenar al personal docente en competencias pedagógicas a nivel de educación superior que les permitan usar e integrar las TIC en los procesos formativos; competencias relacionadas con el uso de recursos digitales en los salones de clase, principalmente asimilar reglas de derechos de *copyright* para la utilización de material didáctico (Sánchez, 2013; Sánchez et al., 2015; Sánchez & Emin-Martínez, 2014).

En **Estados Unidos de América**, se diseñó y desarrolló el juego serio Cisco Packet Tracer, el cual es un juego serio de simulación utilizado para potenciar competencias tecnológicas y de gestión, a través del entrenamiento y el aprendizaje técnico, de forma visual e interactiva desarrollado por *Cisco Systems* para ser utilizado principalmente en la *Cisco Networking Academy* (Cabarkapa, 2015; Frezzo et al., 2009; Janitor et al., 2010).

Se considera un juego serio ya que un grupo de docentes del departamento de Informática de la Universidad Técnica de Cluj-Napoca en Rumania, desarrollaron un módulo que integra *Cisco Packet Tracer* con *Moodle* para potenciar estrategias de e-learning. El módulo denominado “*PTActivity*”, permite integrar plataformas e-learning con herramientas de simulación de red, además, ofrece un *Feedback* automático tanto para el estudiantado como para el profesorado (Petcu et al., 2013).

En **Brasil**, se desarrolló el juego *DEBORAH Game*, el cual fue creado por Edgard Bruno Cornacchione Junior del laboratorio de educación tecnológica en Contabilidad de la Universidad de São Paulo. *DEBORAH Game* es un juego serio que tiene como objetivo potenciar competencias tecnológicas y contables tanto en el profesorado como en el estudiantado (DEBORAH Game, 2017; Malaquias et al., 2018).

Es importante mencionar que se ha realizado un relevamiento de juegos serios utilizados en la formación o adquisición de competencias y habilidades tecnológicas en particular, sin embargo, existen antecedentes de utilización de juegos serios para la formación o adquisición de competencias en otras áreas del conocimiento, tal como el caso de **Holanda** donde se desarrolló el juego serio *Frecuency 1550*, el cual tiene como objetivo la formación de competencias sociales y culturales del país, específicamente todo lo relacionado con la historia medieval de la ciudad de Ámsterdam (Apezteguía et al., 2014; Waag-Society, 2019).

En esta línea, en **Canadá** se ha desarrollado el juego serio *RollerCoaster Tycoon* para la formación en habilidades en gestión, pensamiento estratégico y de planificación (Romero & Turpo, 2012). Otro ejemplo igualmente **canadiense**, es el juego *Spirits of Spring* el cual fue desarrollo concientizar a la población en competencias relacionadas aspectos sociales, éticos y legales del uso de las TIC, en este caso específico, la prevención de la utilización de la tecnología como medio para el *bullying* (OmniumGames, 2014).

En **Finlandia** se ha desarrollado el juego *NoviCraft* el cual tiene como propósito principal potenciar a través del entrenamiento la formación de competencias pedagógicas en los jugadores a través de la utilización y aplicación de las TIC en los procesos formativos, específicamente, dirigido a la formación de competencias relacionadas con el trabajo en equipo, gestión y liderazgo (Häkkinen et al., 2012; Holloway, 2009; Juzeleniene et al., 2014).

En la [Tabla 3.1](#), se resume la información de los 12 juegos serios anteriormente descritos, los cuales han sido clasificados de acuerdo con el país de origen y objetivo principal del juego o bien, intención pedagógica del mismo (competencia para la cual fue inicialmente destinado a potenciar en los usuarios, es decir, se refiere al objetivo caracterizante del juego serio).

*Tabla 3.1 Juegos serios utilizados en procesos de desarrollo de competencias*

País	Juego serio	Objetivo caracterizante del juego serio				
		Tecnológico	Pedagógico	Social	Ético-Legal	Gestión
Argentina	RITA	✓	✓			
	AstroCódigo	✓	✓			
	FraccionAR		✓			
	Desafíate		✓			
Irlanda	GSD Sim	✓	✓			
Francia	Tamagocours	✓	✓		✓	
USA	Cisco Packet Tracer	✓	✓			✓
Brasil	DEBORAH Game	✓	✓			
Holanda	Frecuency 1550		✓	✓		
Finlandia	NoviCraft	✓	✓			✓
Canadá	Spirits of Spring		✓	✓	✓	
	RollerCoaster Tycoon Touch		✓			✓

Dado que la muestra de los juegos serios visibles en la [Tabla 3.1](#) es pequeña, no es posible marcar tendencias ni conclusiones contundentes, pero sí es viable conocer la caracterización global de la muestra seleccionada. Por ello, la inclusión de los porcentajes en la descripción de cada criterio se orienta en este sentido.

- ✓ El 33.33% (4) de los juegos serios analizados fueron diseñados y desarrollados en Argentina, el 16.66% (2) en Canadá y el resto del porcentaje (50%) se divide en proporciones iguales (8.33%) entre los otros países; Irlanda, Francia, USA, Brasil, Holanda y Finlandia, quienes registran un juego cada uno.
- ✓ El 100% de los juegos serios analizados coinciden en que al menos uno de sus objetivos caracterizante es la formación de competencias pedagógicas y, el 58.33% de los juegos además de las competencias pedagógicas, potencian la

formación y adquisición de competencias o habilidades tecnológicas. Luego, en menor proporción, se evidencia la utilización de juegos serios para potenciar otras áreas del conocimiento; sociales, éticas-legales y de gestión.

Finalmente, se evidencia que, los juegos serios han sido utilizados en diferentes países en procesos de formación de competencias tecnológicas.

### **3.5. Conclusiones del capítulo**

En este capítulo se presentó la definición de juegos serios desde el punto de vista de varios autores, posteriormente, se brindó una definición propia del concepto, la cual reúne los principales aspectos que los identifican (aprendizaje y diversión). Es decir, el concepto toma en cuenta el equilibrio entre lo pedagógico y lúdico, con el propósito de asegurar la jugabilidad e inmersión del jugador. Luego, se realizó una caracterización de los juegos serios a nivel general y, en particular, se brindó una serie de características funcionales. Entre las principales características se destaca que un juego serio debe poseer la capacidad de provocar en el jugador sensaciones y experiencias nuevas (motivación y colaboración) mientras permiten la adquisición o formación de competencias y habilidades en un área del conocimiento determinadas acorde a la intención pedagógica para el cual fue diseñado.

Seguido, se citaron y describieron experiencias en diferentes países en los cuales se han utilizado juegos serios para potenciar competencias tecnológicas y, otras habilidades de distintas áreas del conocimiento. Como resultado se pudo constatar que se registra variedad de experiencias de utilización de juegos serios para la formación de competencias (12 experiencias en 8 países), tanto en América (Argentina, USA, Brasil, Canadá) como en Europa (Irlanda, Francia, Holanda, Finlandia).

# CAPÍTULO IV. CLOUD COMPUTING

#### 4.1. Introducción

Este capítulo presenta la conceptualización de *cloud computing* que es abordada desde el punto de vista de varios autores. Inicialmente, se define y describe el concepto de *cloud computing*. Posteriormente, se brinda una definición propia del término. Seguido, se describen las características esenciales del modelo. Luego, se detallan los modelos de servicios y despliegue ofrecidos en *cloud computing*. Por último, se abordan las posibilidades de *cloud computing* en educación.

Los contenidos planteados a desarrollar en este capítulo permiten brindar respuesta a las preguntas de investigación; Plx6, Plx7, Plx8 establecidas para el estudio y enumeradas en el Capítulo I:

Plx6 ¿Cómo se define o describe el concepto de *cloud computing*?

Plx7 ¿Cuáles son las características esenciales de *cloud computing*?

Plx8 ¿Cómo se describen los modelos de servicio y despliegue en *cloud computing*?

De acuerdo con lo señalado por Sandí & Bazán (Sandí-Delgado & Bazán, 2017, 2019), los avances tecnológicos ocasionan que cada día el volumen de datos e información crezca de forma acelerada, por tanto, las organizaciones, empresas, el comercio, el gobierno y, en general, la sociedad, han tenido que buscar nuevos mecanismos que garanticen el respaldo y la seguridad de la información así como la disponibilidad y fácil acceso a la misma.

En esta línea, se realiza el abordaje del concepto de *cloud computing* como un nuevo modelo computacional que ofrece de forma compartida una serie de servicios y recursos tecnológicos de alto nivel, los cuales se ofrecen de forma pública o privada, bajo demanda y con alta disponibilidad, que pueden ser accedidos por los usuarios mediante un browser con conexión a internet (Sandí-Delgado & Bazán, 2017, 2019).

#### 4.2. Conceptualización y características de *cloud computing*

El concepto de *cloud computing* o computación en la nube ha sido definido ampliamente por distintos autores expertos en el área (Bazán et al., 2017; Jula et

al., 2014; Mell & Grance, 2010; Murazzo et al., 2010, 2015; Nieto & Bazán, 2013; Rodríguez, Murazzo, et al., 2014; Rodríguez, Valenzuela, et al., 2014; Sandí-Delgado & Bazán, 2019; Shojaei et al., 2018; Velásquez et al., 2018; Velurtas & Bazán, 2018), la definición del concepto puede variar de un autor a otro dependiendo del foco de interés y el grado de experticia de cada uno.

Es importante indicar que la definición que ha sido destacada en las referencias analizadas es la ofrecida por Mell & Grance (2010) investigadores del Instituto Nacional de Estándares y Tecnología (NIST, *National Institute of Standards and Technology*), la cual se presenta a continuación:

**Según** (Mell & Grance, 2010, p. 2)

*Cloud computing* es un nuevo modelo de servicios que permite el acceso a la red omnipresente y conveniente a un conjunto compartido de recursos computacionales configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que se pueden **aprovisionar y liberar rápidamente** con un esfuerzo mínimo de gestión por parte del proveedor de los servicios. **Los usuarios pueden utilizar el servicio en cualquier momento y en cualquier lugar a través de Internet**, directamente a través del navegador.

Con el propósito de ampliar aún más el concepto de *cloud computing*, se presentan otras definiciones acotadas por investigadores expertos en el área:

**Según** (Murazzo et al., 2015, p. 523)

Es un modelo para permitir el acceso en la red omnipresente, conveniente y **bajo demanda** a un conjunto compartido de recursos informáticos configurables tales como, redes, servidores, almacenamiento, aplicaciones y servicios que pueden ser **rápidamente aprovisionados y liberados** con una mínima gestión, esfuerzo o interacción con el proveedor o administrador del servicio. La característica básica de este modelo es que **los recursos y servicios informáticos son ofrecidos y consumidos bajo demanda**, como servicios a través de la Internet.

**Según** (Bazán et al., 2017, p. 108)

La Computación en la nube (*Cloud Computing*) es un paradigma que posibilita el **acceso ubicuo bajo demanda a servicios TIC** accesibles a través de Internet. Se

sustenta en la idea **consumir sólo lo que se necesita y pagar por lo que se consume**, de una manera similar a los servicios de luz o de gas.

**Según** (Nieto & Bazán, 2013, p. 7)

Modelo de computación que permite **demandar a través de internet recursos compartidos** como procesamiento en servidores, almacenamiento, aplicaciones y servicios, los cuales **pueden ser rápidamente aprovisionados y liberados** sin ningún esfuerzo de administración por parte del proveedor de estos servicios.

**Según** (Rodríguez, Valenzuela, et al., 2014, p. 39)

Se puede pensar a *Cloud Computing* como un modelo de aprovisionamiento de recursos IT que potencia la prestación de servicios IT y servicios de negocio, facilitando la operativa del usuario final y del prestador del servicio. La característica básica de este modelo es que **los recursos y servicios informáticos**, tales como infraestructura, plataforma y aplicaciones, **son ofrecidos y consumidos como servicios a través de la Internet** sin que los usuarios deban tener ningún conocimiento de lo que sucede detrás.

**Según** (Salas-Zárate & Colombo-Mendoza, 2012, p. 48)

La computación en la nube es un nuevo estilo de computación que es **dinámicamente escalable** y, a menudo, **se proporcionan recursos virtualizados a través de Internet**.

A partir de las definiciones anteriormente acotadas y, en sus características comunes identificadas, se ofrece nuestra propia definición para *cloud computing*. Así, la definición adoptada para esta investigación consiste en que:

*Cloud computing* se define como un nuevo modelo computacional que **ofrece de forma compartida** una serie de **servicios y recursos tecnológicos de alto nivel** (infraestructuras, plataformas y aplicaciones), los cuales podrían **brindarse de forma pública o privada, bajo demanda, de elasticidad rápida y con alta disponibilidad**, que pueden ser accedidos por los usuarios

mediante un browser con conexión a internet y, usualmente, **de pago por uso** (consumir sólo lo que se necesita y pagar por lo que se consume).

Con esta definición de *cloud computing*, se atiende a las características principales identificadas en la revisión de literatura y se ofrece un concepto que permite comprender con mayor amplitud el modelo *cloud computing*, además, podría impactar de forma positiva en el momento de definir una metodología y arquitectura de software para la producción de juegos serios orientados a promover la adquisición de conocimientos y habilidades relacionadas con las competencias tecnológicas.

El modelo de nube enunciado por la NIST responde a la metáfora 5-3-4, es decir, se destaca o identifica por 5 características esenciales, 3 modelos de servicio y 4 modelos de despliegue (Bazán, 2015; Bazán et al., 2017; Mell & Grance, 2010; Salas-Zárate & Colombo-Mendoza, 2012), características y modelos que se describen en detalle en las siguientes secciones.

#### 4.2.1. Características esenciales de *cloud computing*

Acorde a lo señalado por varios investigadores (Bazán, 2015; Joyanes, 2009, 2012; Mell & Grance, 2010; Salas-Zárate & Colombo-Mendoza, 2012; Sandí-Delgado & Bazán, 2017) el modelo de nube enunciado por NIST se destaca por las siguientes cinco características:

- ✓ **Amplio acceso a la red (*Broad network access*)** - Las capacidades están accesibles en la red a través de mecanismos estándar que permiten el acceso desde plataformas cliente-heterogéneas.
- ✓ **Elasticidad rápida (*Rapid elasticity*)** - Las capacidades son aprovisionadas y liberadas rápida y elásticamente (y en algunos casos automáticamente).
- ✓ **Servicio medido (*Measured service*)** - Los sistemas *cloud computing* controlan y optimizan automáticamente el uso de los recursos proporcionando alguna capacidad de medición (habitualmente pago por uso o cargo por uso).

- ✓ **Autoservicio a demanda (*On-demand self-service*)** - Un consumidor puede aprovisionar unilateral y automáticamente recursos de computación según necesite.
- ✓ **Recursos mancomunados (*Resource pooling*)** - Los recursos de computación de los proveedores son agrupados para servir a múltiples clientes usando un modelo *multi-tenant* (multi-inquilino).

En la próxima sección se describe el modelo de servicios enunciado por NIST para *cloud computing*.

#### 4.3. Modelos de servicio en *cloud computing*

Los tres modelos de servicio enunciados por NIST para *cloud computing* se describen acorde a diferentes autores que los analizan y describen (Mell & Grance, 2010; Murazzo et al., 2010, 2015; Nieto & Bazán, 2013; Sandí-Delgado & Bazán, 2017, 2019; Shojaei et al., 2018).

- ✓ **Infraestructura como servicio (IaaS, *Infrastructure as a Service*)**. Es un modelo de servicio que ofrece los recursos de computación (procesamiento, servidores, el almacenamiento y la red) en los cuales se ejecuta el software, incluyendo aplicaciones y sistemas operativos. El proveedor tiene el control absoluto de la infraestructura subyacente de la nube. El consumidor controla los sistemas operativos, el almacenamiento y las aplicaciones desplegadas, en ocasiones la red (firewalls). Entre las características principales identificadas en el modelo IaaS están (Bazán et al., 2017, p. 113):
  - Elasticidad dinámica de los recursos. Cuando los recursos no son requeridos, se liberan de manera dinámica y transparente.
  - Provisión de recursos bajo demanda. Los recursos son asignados a medida que los necesita quien está desplegando su entorno de TI en la nube.
  - Uso eficiente de los recursos. Para que los recursos brinden el máximo de rendimiento, se utilizan técnicas de virtualización y arquitectura *multi-tenant*.

- Modelo de pago por uso. Este modelo permite que la TI se ofrezca como servicio e implica establecer métodos y criterios para poder medir el uso de TI.
- Gestión más eficiente de los *datacenters*. Se obtienen enormes beneficios de la economía de escala, estandarización, se reducen los costos y se hace un uso más eficiente de la energía.

A continuación, se describen las funciones propias que han sido identificadas en el modelo de servicios IaaS (Bazán et al., 2017, pp. 113–114):

- Gestión dinámica de recursos (ejemplo; inicio/parada de máquinas virtuales, creación/eliminación de volúmenes, asignación de mayor capacidad de cómputo).
- Monitorización de recursos (CPU, memoria, ancho de banda).
- Funciones de automatización (ejemplo; balance automático de carga, reglas de autoescalado horizontal, manejo de prioridades en la asignación de recursos).
- Gestión de consumos y costos que permite medir los consumos de recursos de TI para poder cobrar al cliente en función de ellos.

✓ **Plataforma como servicio (PaaS, *Platform as a Service*)**. Es un modelo de servicio que ofrece el desarrollo de aplicaciones y su despliegue como servicios en internet, tanto de aplicaciones propias como adquiridas (desarrolladas usando entornos de programación soportados por el proveedor), permitiendo que se desarrolle aplicaciones de software personalizadas. El consumidor no gestiona ni controla la infraestructura subyacente de la nube, incluida la red, los servidores, los sistemas operativos o el almacenamiento.

Entre los principales tipos de servicios identificados que ofrece el modelo PaaS están (Bazán et al., 2017, p. 114):

- Desarrollo de extensiones SaaS. Proveedores de SaaS definen APIs para construir sobre ellos (Ejemplo: *SalesForce, Facebook*).

- Plataformas propietarias. Algunos solo despliegan otros soportan todo el ciclo de vida del software: *Amazon Elastic BeansTalk* (despliegue de aplicaciones en Amazon EC2), *Microsoft Azure* (plataforma .NET), *Google AppEngine* (APIs propias).
- *Open Platform*. Ofertas que se basan en tecnologías abiertas para evitar la dependencia del vendedor (en inglés, vendor lock-in). Son arquitecturas extensibles en las que se puede escoger el entorno de desarrollo, el lenguaje, la base de datos, el servidor de aplicaciones. Ejemplos: *CloudFoundry* (VMWare), *OpenShift* (Red-Hat), *CumuLogic*, *CloudBees*.

Las prestaciones usuales que han sido identificadas que son ofrecidas en el modelo PaaS son (Bazán et al., 2017, p. 114):

- Gestión del ciclo de vida de las aplicaciones. Esto es muy importante dado que las plataformas de desarrollo basadas en la nube son accedidas por varios usuarios a una misma instancia. Esta instancia, de la misma manera que una instancia de clase en la orientación a objetos es creada, tiene un tiempo de vida y eventualmente es destruida o muere.
- Herramientas de desarrollo colaborativo. El modelo de trabajo en la nube es una actividad netamente colaborativa dado que varios acceden a la misma instancia y se deben aplicar las reglas de control de acceso y versiones típicas de este modelo.
- Herramientas para el desarrollo de interfaces de usuario. Como cualquier entorno de desarrollo tradicional u *on-premise*, se debe contar con componentes específicos para la construcción de interfaces de usuario.
- Integración con *Web Services* (SOAP, REST): *Mashups* (Forma de integración y reutilización. Ocurre cuando de una aplicación web es usada o llamada desde otra aplicación, con el fin de reutilizar su contenido o funcionalidad), composición de servicios. En PaaS es mucho más frecuente y necesario construir aplicaciones como composición de componentes existentes y que puedan interoperar entre ellas aprovechando así un entorno de ejecución global.

- Integración con BD SQL y NoSQL. Al igual que los entornos de desarrollo clásico, se debe contar con manejadores que permitan establecer conexiones con las bases de datos, pero estando estas en un entorno cloud.
  - Mecanismos de extensibilidad (*plugins*) para configurar nuevos *frameworks*.
  - Independencia del proveedor IaaS.
- ✓ **Software como servicio (SaaS, Software as a Service).** Es un modelo de servicio de alto nivel que brinda aplicaciones bajo demanda a usuarios finales, que se ejecutan en una infraestructura de *cloud computing*. SaaS ofrece al consumidor todos los recursos necesarios; red, servidores, sistemas operativos y almacenamiento. SaaS le permite al usuario acceder a una aplicación a través de internet, la cual reside y se ejecuta desde los servidores de un proveedor SaaS, la cual puede ser utilizada de forma gratuita o bien, por el tiempo y recursos consumidos, según sea el acuerdo establecido entre el proveedor y el consumidor. El usuario no tiene el privilegio de gestionar o controlar la infraestructura subyacente de la nube, sin embargo, puede controlar la configuración personal de las aplicaciones. Los beneficios que identifican principalmente el modelo SaaS son (Bazán et al., 2017, p. 115):
- No hay costos de licencia.
  - La gestión de TI se simplifica.
  - Las actualizaciones son automáticas y constantes.
  - Ubicuidad (accesible desde cualquier dispositivo).
  - Integración a través de APIs.

Así mismo, se evidencian desventajas para el modelo de servicios SaaS, tales como:

- Dependencia de la red.
- Adaptabilidad (En el sentido que las aplicaciones desplegadas en la nube no se adaptan a los requisitos específicos de cada usuario, sino que brindan características universales en las se sustentan el modelo *multi-tenant*).
- Aspectos legales y de seguridad (se refiere específicamente a que se modifica el modelo “acuerdo de nivel de servicio” (SLA, *Service Level*

*Agreement)* dado que se definen nuevas responsabilidades propias de un modelo de servicio diferente.

- Dependencia del vendedor (En términos de que cual es la plataforma donde se alojan las aplicaciones en SaaS y la incertidumbre de su disponibilidad en el tiempo).
- Costos: para PYMES es ventajoso.

#### 4.4. Modelos de despliegue en *cloud computing*

Los modelos de despliegue de *cloud computing* han sido ampliamente descritos por diferentes autores (Bazán, 2015; Bazán et al., 2017; Jula et al., 2014; Mell & Grance, 2010; Murazzo et al., 2015; Salas-Zárate & Colombo-Mendoza, 2012; Sandí-Delgado & Bazán, 2017, 2019), los cuales se citan y resumen a continuación:

- ✓ **Nube pública (*Public cloud*)** - Ofrece servicios de computación virtualizados externos al cliente en la cual la infraestructura se encuentra al servicio del público o empresas en general, los cuales son propiedad de la organización que vende los servicios.
- ✓ **Nube privada (*Private cloud*)** - Ofrece servicios de computación virtualizados y de manera exclusiva para una organización con múltiples clientes (Ejemplo, departamentos de una empresa). Puede ser de su propiedad (*on-premise*), alquilada (*off-premise*) o una combinación de ambas. Permite una gestión flexible y ágil de los recursos de la organización. Los principales proveedores públicos de las IaaS comenzaron creando nubes privadas para mejorar la gestión de sus *datacenters*. La gestión e infraestructura interna es operada o controlada sólo por la organización que ofrece dicho servicio.
- ✓ **Nube híbrida (*Hybrid cloud*)** - Ofrece servicios de infraestructura de tipo mixto, es decir, una combinación de nube privada y pública que permite gestionar los picos de carga obteniendo recursos de la nube pública. El uso de una nube pública es totalmente transparente para usuarios de la nube privada. La infraestructura *cloud computing* está compuesta por dos o más nubes;

privadas o públicas, las cuales permiten enviar datos o aplicaciones entre ellas. Siguen siendo entidades únicas, pero están unidas entre sí por tecnologías estandarizadas o propietarias que permiten la portabilidad de datos y aplicaciones.

- ✓ **Nube comunitaria (*Community cloud*)** - Ofrece servicios comunes a clientes similares donde la infraestructura *cloud computing* es compartida por varias organizaciones y cada componente conserva su autonomía. La relación entre componentes se hace mediante tecnologías (propietarias o estandarizadas) que permitan la portabilidad de datos y aplicaciones. Puede ser propiedad de una o más organizaciones, alquilada o combinación de ambas.

En la [Figura 4.1](#), se puede observar con facilidad la integración de los modelos de servicio de *cloud computing*, en el cual IaaS se convierte en un modelo de servicios indispensable (requisito) para contar con alguno de los otros modelos de servicios. Además, se puede apreciar que se puede obtener o adquirir el modelo de servicios SaaS soportado directamente sobre IaaS, es decir, se podría desplegar SaaS sobre IaaS aunque la plataforma de desarrollo no sean “aaS” (Bazán et al., 2017).

Finalmente, Bazán et al (2017, p. 121) señalan que la virtualización de la nube ha potenciado el surgimiento de nuevos niveles donde: “Los conceptos de SaaS (*Software as a Service*), PaaS (*Platform as a Service*) e IaaS (*Infrastructure as a Service*), hacen de la red un recurso escalable y constituyen un medio para desarrollo y despliegue de aplicaciones globales”. Es decir, que además del desarrollo de aplicaciones, *cloud computing* posibilita el despliegue, mantenimiento e interoperabilidad de estas.

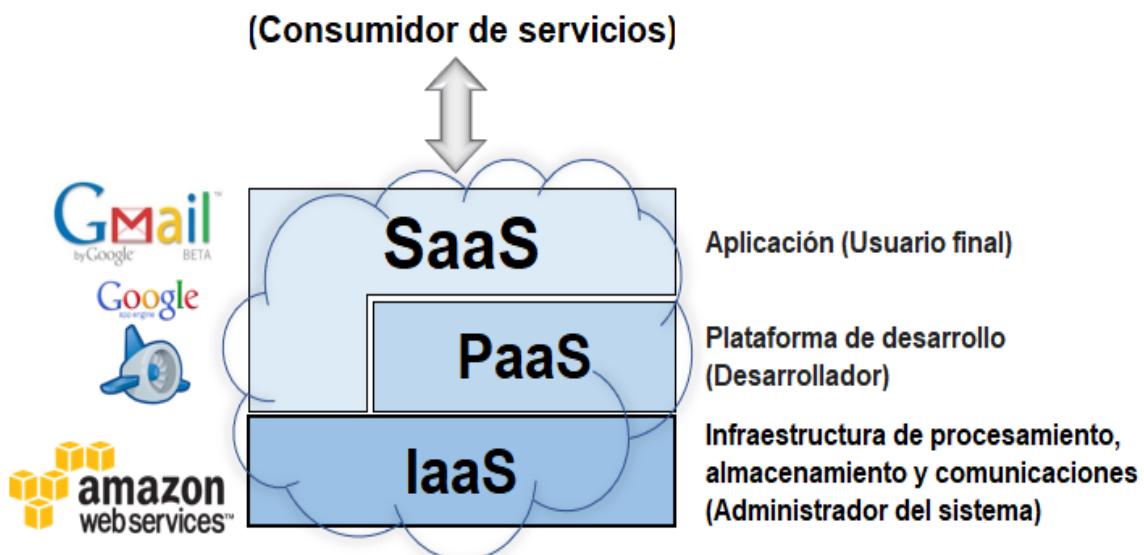


Figura 4.1 Integración de los modelos de *cloud computing*

Fuente: Elaboración propia a partir del texto de Bazán et al (2017, p. 117)

La Figura 4.2, muestra el modelo de nube propuesta por NIST, el cual integra las 5 características esenciales, los 3 modelos de servicio y los 4 modelos de despliegue de *cloud computing*.

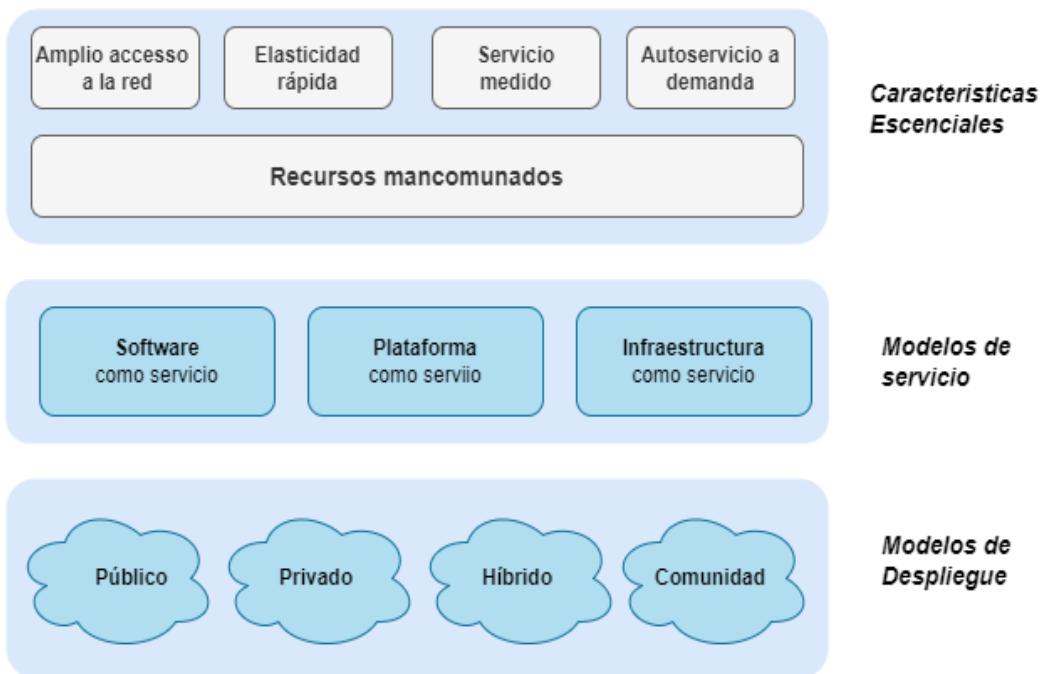


Figura 4.2 Modelo de nube integrado propuesto por NIST

Fuente: Elaboración propia a partir del texto de Bazán et al (2017, p. 117)

#### 4.5. Posibilidades de *cloud computing* en educación

De acuerdo con lo expuesto por Sandí & Bazán (Sandí-Delgado & Bazán, 2019), la gestión del conocimiento; misma entendida como el proceso o práctica que permite capturar, crear, almacenar experiencias de aprendizaje para transferir o compartir con otros, se vería fortalecida a través del uso e integración de *cloud computing*, ya que podría potenciar el rendimiento académico, la eficacia y la eficiencia de la comunidad universitaria en general, al brindar y facilitar a las personas usuarias el acceso, edición colaborativa y sincronización de sus materiales de referencia digitales en cualquier momento, desde cualquier lugar y al utilizar cualquier dispositivo con conexión a internet. Al respecto, Sandí & Bazán (Sandí-Delgado & Bazán, 2019) señalan que el componente colaborativo que ofrece *cloud computing* es vital en los procesos formativos, debido a que desde el punto de vista educativo, la colaboración permite que dos o más individuos comparten vivencias, experiencias, puntos de vista diferentes, se generan intercambios de opiniones, materiales, discusiones y, principalmente, se potencia el análisis grupal, donde se ve reflejado el aporte crítico de cada participante en la construcción conjunta de nuevos conocimientos; a partir de las experiencias y vivencias propias de cada participante, es decir, se facilita un *feedback* (Cruz-Alvarado et al., 2017; Sandí-Delgado & Cruz-Alvarado, 2016).

#### 4.6. Conclusiones del capítulo

En este capítulo se presentaron diferentes definiciones de *cloud computing*, las cuales permitieron identificar una serie de aspectos que caracterizan a este modelo computacional, mismas que posteriormente se toman como referencia para la elaboración de una definición propia de *cloud computing*. Luego, se identificaron y describieron las 5 características esenciales de este modelo, mismas que van desde el acceso a la red, hasta los recursos mancomunados, pasando por elasticidad rápida, servicio medio y servicios bajo demanda. Además, se describe a detalle los 3 modelos de servicios (IaaS, SaaS, PaaS) y los 4 modelos de despliegue ofrecidos por *cloud computing* (nube pública, privada, hibrida, comunitaria), modelos que permiten hacer de la red un recurso escalable y, contribuyen al desarrollo, mantenimiento y despliegue de diferentes aplicaciones. Por último, se describe las posibilidades de *cloud computing* en educación, donde se resalta que, este modelo

computacional podría potenciar el rendimiento académico, la eficacia y la eficiencia, a través de la potencialización de la colaboración y la implementación del *feedback*.

# **CAPÍTULO V.**

# **CONSIDERACIONES**

# **TEÓRICAS EN EL**

# **DESARROLLO DE SOFTWARE**

#### 4.1. Introducción

En este capítulo se presentan y describen aspectos teóricos a considerar en el desarrollo de software a nivel general, aspectos que son abordados desde el punto de vista de varios autores. Se inicia con la descripción de los lenguajes de programación, en el cual se aborda y describe brevemente los paradigmas y tipos de lenguajes. Luego, se abordan los procesos de software (especificación, desarrollo, validación y evaluación), así como los modelos de desarrollo (cascada, proceso incremental, prototipos, espiral, desarrollo de aplicación rápida y el ciclo de vida ágil) y el diseño centrado en el usuario. Seguido, se describe el proceso de ingeniería de requerimientos, y se aborda los dos tipos de clasificación de requerimientos (funcionales y no funcionales). Posteriormente, se presenta el proceso de diseño del software, dentro del cual se consideran aspectos relacionados con el diseño y estilo arquitectónico y los diagramas de flujo de datos. Además, se aborda la etapa de la codificación de software, en el cual se describe brevemente aspectos relacionados con las características del código y la reutilización del software. Por último, se hace referencia al proceso de las pruebas de software y a las técnicas utilizadas para tal propósito.

Los contenidos planteados a desarrollar en este capítulo, permite brindar respuesta a la pregunta de investigación; Plx9 establecida para el estudio y enumerada en el Capítulo I:

Plx9 ¿Qué aspectos se deben considerar en el desarrollo de software?, ¿Cómo se describen o caracterizan estos aspectos?

Según varios autores (Melegati et al., 2019; Prasad & Verma, 2016; Sommerville, 2016) durante el desarrollo de software se consideran varios aspectos, los cuales van desde la definición de lenguajes de programación, análisis de requerimientos, diseño, desarrollo, verificado, entre otros, hasta llegar al producto final, en este caso la implementación del software. Aspectos que se enumeran con mayor precisión en las próximas secciones.

Es importante recordar y resaltar que en la ingeniería del software se debe considerar y comprender que en el desarrollo de diferentes tipos de software se puede recurrir a diversas técnicas de ingeniería.

## 4.2. Lenguajes de programación

Los lenguajes de programación consisten en una serie de instrucciones que pueden ser procesadas por una computadora para realizar una acción u operación en específico y, en una secuencia dada (IEEE Computer Society, 1990).

Por tanto, Prasad & Verma (2016) señalan que para diseñar un sistema informático se requiere que los programadores determinen al menos 3 tareas:

- ✓ Las instrucciones por realizar.
- ✓ El orden en que deben realizarse esas instrucciones.
- ✓ Los datos necesarios para realizar esas instrucciones.

### 4.2.1. Paradigmas de programación

Los paradigmas de programación indican el método para la construcción de un programa de computación, así como organizar las tareas que debe llevar a cabo un programa (INET, 2016; Vaca, 2012). Los paradigmas definen los estilos de programación, los cuales pueden resolver problemas de diferentes formas.

De acuerdo con Prasad & Verma (2016), la programación se puede clasificar en tres categorías: programación no estructurada, programación estructural y programación orientada a objetos.

- ✓ **Programación no estructurada** - Este tipo de programación consiste en la escritura de programas que contienen un único programa principal, es decir, todas las acciones de como las entradas, procesamiento y salidas se realizan desde un solo programa (Prasad & Verma, 2016).
- ✓ **Programación estructural** - Es una técnica diseñada para mejorar la confiabilidad y claridad de los programas. En ellas se integran técnicas de programación que incluyen estructuras de código de secuencia, selección y repetición (Hunt, 1979; Vinhas, 2016).

- ✓ **Programación Orientada a Objetos** - Consiste en un estilo de programación que permite crear objetos que interactúan entre sí. Además, permite que piezas de código se reutilicen e intercambien entre distintos programas. Este estilo de programación se enfatiza en diferentes conceptos como lo son: objetos, clases, abstracción, herencia y polimorfismo (Ducasse & Pollet, 2017; Lafore, 2001).

#### 4.2.2. Tipos de lenguajes de programación

En los primeros años de la programación de las computadoras, cada una de las instrucciones fue ejecutada de forma binaria, agregando 1 o 0. Sin embargo, con el pasar de los años los programas fueron creciendo y resultaba muy difícil recordar todas las instrucciones, por tanto, se inició una etapa en la que se emplearon otras técnicas y con ello, el surgimiento de nuevos idiomas que pudieran ser entendidos por las computadoras y fueran más fácil de aprender para los humanos. De acuerdo con Prasad & Verma (2016) los lenguajes de programación se dividen en tres grandes categorías:

- ✓ **Lenguaje de máquina** - Es el lenguaje nativo de las computadoras, el lenguaje empleado por la Unidad Central de Procesamiento y es el quién emplea 0 y 1 para representar las instrucciones (Lee, 2017).
- ✓ **Lenguaje ensamblador** - Es un lenguaje prescriptivo, es decir, deben ser programas pensados en términos de la maquina con instrucciones simbólicas empleando letras en lugar de 0 y 1, y que resuelven problemas de forma secuencial (Lee, 2017).
- ✓ **Lenguaje de alto nivel** - Es un lenguaje que permite crear programas de forma más descriptiva. Emplea conjuntos de palabras y símbolos que son utilizados como reglas para brindar las instrucciones. Estos programas deben ser transformados por compiladores para que las computadoras los puedan reconocer (Lee, 2017).

### 4.3. Procesos de software

Los procesos de software consisten en todas aquellas actividades relacionadas que guían a la producción de un sistema de software (Sommerville, 2016). Es importante indicar que existen diferentes tipos de sistemas de software, por tanto, los procesos no son únicos, cada uno de ellos responde a diferentes variables, como la organización o compañía que lo desarrolla, requerimientos del público meta y las destrezas de quien escribe el software.

No obstante, para Sommerville (2016, p. 43) existen 4 procesos que deben ser aplicados en todo proyecto que conduzca a la producción de un sistema de software:

- ✓ **Especificación de software** - Se debe definir la funcionalidad del software y las restricciones sobre su funcionamiento.
- ✓ **Desarrollo de software** - El software para cumplir con la especificación debe ser producido.
- ✓ **Validación del software** - El software debe validarse para garantizar que hace lo que el cliente desea.
- ✓ **Evolución del software** - El software debe evolucionar para satisfacer las necesidades cambiantes de los clientes.

A pesar de no existir un proceso universal para el desarrollo de sistemas de software, muchas organizaciones y empresas emplean técnicas o prácticas basadas en modelos de ingeniería para mejorar los procesos en el desarrollo de software.

#### 4.3.1. Modelos de desarrollo de software

Los modelos de desarrollo de software también pueden ser llamados ciclo de vida del desarrollo de software, el cual representa de forma simplificada un proceso de software. A continuación, se presentan 5 modelos para el desarrollo de software:

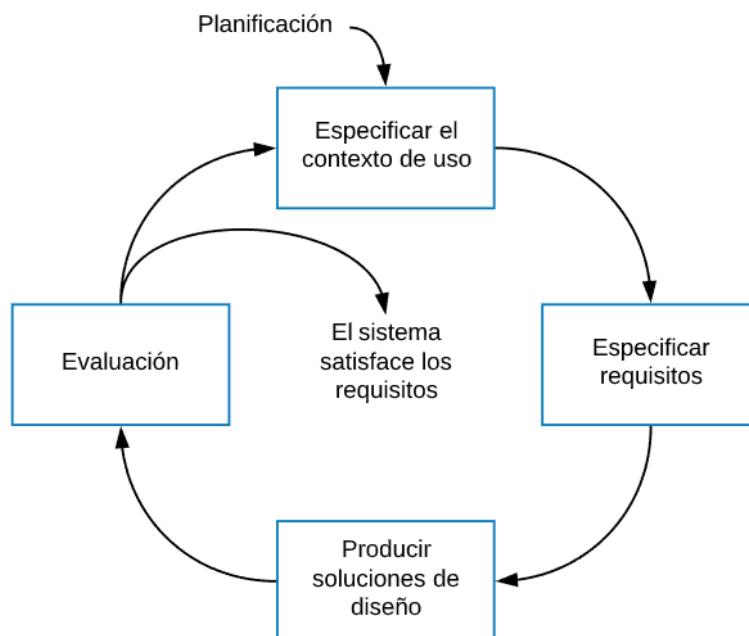
- ✓ **Modelo cascada** - Este modelo es el más antiguo y reconocido de todos. Ha sido empleado desde 1970 y tiene como característica principal que sus pasos se ejecutan de forma secuencial (Lozano, 2017). Da inicio con las fases de análisis de requisitos, luego continua con el diseño, codificación, pruebas y finaliza con mantenimiento. Cada etapa inicia y finaliza antes de comenzar la siguiente etapa (Alshamrani & Bahattab, 2015).
- ✓ **Modelo de proceso incremental** - Este modelo consiste en una evolución del modelo de cascada, tiene como objetivo mejorar la lentitud del ciclo de vida del desarrollo del software (Isaias & Issa, 2015). El modelo presenta avances, los cuales son llamados incrementos que de forma progresiva otorgan mayor funcionalidad al sistema conforme se brindan entregas al cliente (Lozano, 2017).
- ✓ **Modelo de prototipos** – Este modelo consiste en un marco iterativo en donde se tiene como idea crear una parte o totalidad del sistema de software en una versión piloto. Se enfoca más en la creación del software centrado en el usuario y menos en la documentación (Isaias & Issa, 2015). Este modelo tiene como característica principal reducir el riesgo de producir sistemas de software que no satisfacen en su totalidad al cliente, sin embargo, tiende a confundir a los usuarios, podrían creer que ya se cuenta con un producto terminado (Lozano, 2017).
- ✓ **Modelo espiral** - Este modelo también es denominado metamodelo, el cual también puede ser utilizado por otros modelos (Alshamrani & Bahattab, 2015). Combina elementos del modelo de cascada y prototipos, y está centrado en la evaluación de riesgos. Tiene como fundamento realizar entregas del producto cada vez más completa que la anterior, siempre teniendo presente los riesgos que esto puede causar al proyecto. La espiral se divide en 4 segmentos: decisión de objetivos, evaluación y reducción de riesgos, desarrollo y evaluación, y planificación (Lozano, 2017).

- ✓ **Modelo de desarrollo de aplicación rápida** - Este modelo tiene como enfoque realizar de forma rápida la entrega del sistema de software (Sommerville, 2016). Además, el modelo de desarrollo de aplicación rápida contiene herramientas que facilitan las implementaciones dentro de un tiempo predefinido (Isaias & Issa, 2015). Dispone de un proceso lineal secuencial en donde se divide el proyecto en módulos para que sean desarrollados por cada equipo de trabajo, y se integran al final del proyecto (Lozano, 2017).
- ✓ **Modelo de ciclo de vida ágil** - Es un modelo con un enfoque alternativo que busca contrarrestar la rigidez y falta de flexibilidad de otros modelos de desarrollo. Este modelo, contiene las mejores características de otros modelos y los integra en uno solo (Isaias & Issa, 2015). Este modelo contiene prácticas que permiten el desarrollo del software más eficiente y ligera, de forma iterativa e incremental (Anwer et al., 2017). Finalmente, de acuerdo con Fowler & Highsmith (2001) el manifiesto del modelo ágil se basa en 12 principios; 1) Satisfacción del cliente mediante la entrega temprana y continua de software viable. 2) Los requisitos cambiantes son bienvenidos incluso en etapas posteriores de desarrollo. 3) Colaboración y comunicación frecuente entre clientes y desarrolladores. 4) Entrega frecuente de software de trabajo. 5) Apoyar y motivar a personas confiables involucradas en el desarrollo de software. 6) Utiliza la comunicación cara a cara. 7) El software de trabajo es la principal medida de progreso. 8) Se mantiene un ritmo constante a través del desarrollo sostenible. 9) Presta atención al buen diseño continuamente. 10) Mantener las cosas simples. 11) Los equipos autoorganizados pueden desarrollar una mejor arquitectura, requisitos y diseño. 12) El equipo regularmente refleja cómo ser más efectivo.

#### 4.3.2. Diseño Centrado en el Usuario

El Diseño Centrado en el Usuario (DCU) o bien, (UCD, por sus siglas del inglés *User Centered Design*), es un enfoque de diseño en el cual el proceso está dirigido por la información vinculada con las personas que utilizarán el producto por

diseñarse (Hassan-Montero & Ortega-Santamaría, 2009), es decir, se da un proceso cíclico donde los usuarios influyen sobre las decisiones de diseño y sobre los objetivos pretendidos por alcanzar con el producto, asimismo, la evaluación del producto y la usabilidad del diseño son realizadas de forma iterativa y mejorada de forma incremental (Evans et al., 2016). La [Figura 5.1](#) resume el modelo DCU.



**Figura 5.1** Modelo del diseño de procesos centrados en el usuario.

Fuente: Elaboración propia a partir del texto de (Evans et al., 2016; Marfisi-Schottman et al., 2010).

Otros autores (Elnaggar & Reichardt, 2016; Garita et al., 2019; Massa et al., 2017) agregan que la filosofía principal del DCU se relaciona con otras metodologías y técnicas que se enfocan en las necesidades, limitaciones y los requerimientos de los usuarios. Además, los mismos autores señalan que el modelo DCU se rige por los siguientes principios:

- ✓ Los usuarios finales representativos deben participar activamente de manera temprana y continua durante todo el desarrollo.
- ✓ El desarrollo debe experimentar muchos ciclos iterativos para llegar a los requisitos de los usuarios finales.
- ✓ Los prototipos deben crearse temprana y continuamente para ayudar a visualizar y evaluar ideas.

- ✓ El proceso de desarrollo debe ser realizado por equipos interdisciplinarios.

#### 4.4. Ingeniería de requerimientos

La ingeniería de requerimientos consiste en el proceso encargado de satisfacer las necesidades del cliente a través de un conjunto de técnicas y procedimientos que permitan definir un proyecto de software (Gómez-Fuentes, 2011).

Este proceso proporciona a los ingenieros las características de operación del sistema, es decir, se encarga de indicar “**que**” debe hacer el sistema, las funciones y el comportamiento que se desea. Estos requerimientos permiten que los encargados del proyecto y el cliente puedan valorar el alcance y cumplimiento de los requerimientos una vez que finalice el proyecto.

En esta misma línea, la (IEEE Computer Society, 1990) en el estándar 610.121990 señala que los requerimientos se deben entender como una condición que los usuarios necesitan para cumplir con un objetivo o, bien, una capacidad que un sistema debe poseer para satisfacer un contrato o especificación.

Por ello, realizar una buena especificación de requerimientos es determinante en la calidad del software (Aurum & Wohlin, 2005), debido a que, si no se realiza una buena especificación de requisitos, el resultado podría no cumplir con las expectativas de los clientes (Alshazly et al., 2014). Regularmente, los requerimientos se clasifican en dos tipos; funcionales y no funcionales.

- ✓ **Requerimientos funcionales** - Consisten en declaraciones de los servicios que un sistema debe proporcionar. Entre esas acciones indica que debe hacer y no hacer el sistema en escenarios particulares (Sommerville, 2016).
- ✓ **Requerimientos no funcionales** - Estos requerimientos consisten en restricciones de los servicios que ofrece el sistema. Estos pueden ser la fiabilidad, tiempos de respuesta o capacidad del sistema (Sommerville, 2016).

Los requerimientos funcionales y no funcionales son fundamentales para la asignación de significados semánticos en cada una de las tareas de los sistemas informáticos.

En la ingeniería de requerimientos existen 3 actividades claves que componen los procesos del ciclo de vida relacionada con los requisitos (Aurum & Wohlin, 2005). Estas actividades consisten en los “*requisitos de descubrimiento al interactuar con las partes interesadas (obtención y análisis); convertir estos requisitos en una forma estándar (especificación); y comprobar que los requisitos realmente definen el sistema que el cliente desea (validación)*” (Sommerville, 2016, p. 111).

#### 4.5. Diseño del software

El diseño de software consiste en una actividad en donde se analizan los requisitos para generar una estructura que sirva como base para la construcción de este. Es decir, el diseño tiene como finalidad describir “**como**” se van a lograr las funciones del software mediante una especificación de propiedades de objeto (Ralph & Wand, 2007)

El diseño no solo abarca las especificaciones de los requisitos, sino que también todas aquellas actividades implicadas en la conceptualización, encuadre, implementación, puesta en marcha y modificación del software (Freeman & Hart, 2004; Medvidovic & Taylor, 2010).

##### 4.5.1. Decisiones del diseño arquitectónico

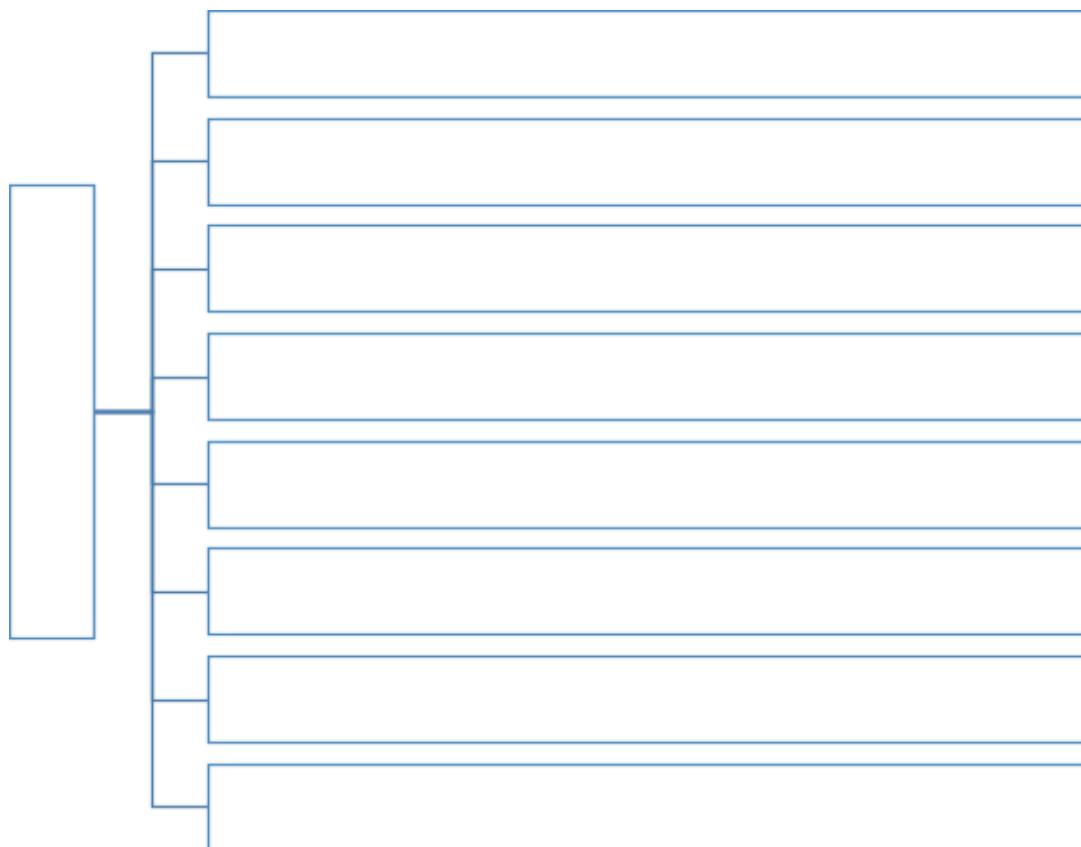
El diseño arquitectónico debe definir una colección de componentes de hardware, software e interfaces del cual se pueda crear un marco para el desarrollo del software (IEEE Computer Society, 1990). Además, se debe presentar una definición de la relación de cada uno de los componentes.

Asimismo, (Jansen & Bosch, 2005) definen las decisiones del diseño arquitectónico como “*una descripción del conjunto de adiciones arquitectónicas, sustacciones y modificaciones a la arquitectura del software, la justificación y las reglas de diseño, las restricciones de diseño y los requisitos adicionales que (parcialmente) realizan uno o más requisitos en una arquitectura*” (p. 2).

Por consiguiente, se puede decir que el principal objetivo del diseño arquitectónico se basa en establecer un diseño con las capacidades de satisfacer los requisitos funcionales y no funcionales.

Para (Sommerville, 2016) no existe un proceso de actividades para el diseño arquitectónico, ya que este depende de múltiples factores, como el tipo de sistema, antecedentes, requisitos específicos y la experiencia del diseñador o arquitecto del sistema. Del mismo modo, el autor agrega que el diseñador debe considerar una serie de interrogantes y tomar en cuenta los requerimientos no funcionales, ya que de estos depende el diseño arquitectónico.

En la [Figura 5.2](#), se pueden observar las preguntas propuestas por el autor.



**Figura 5.2** *Interrogantes a considerar en el diseño arquitectónico*  
Fuente: Elaboración propia a partir del texto de (Sommerville, 2016)

Por tanto, cualquier decisión que se considere en un diseño arquitectónico se va a ver reflejado dentro de los procesos de construcción del sistema de software. Es decir, la arquitectura de software de un sistema es el resultado de la toma de un

conjunto de decisiones de diseño arquitectónico durante un largo periodo de tiempo (Jansen & Bosch, 2005).

#### 4.5.2. Estilos arquitectónicos

Según (Garlan & Shaw, 1994) los estilos arquitectónicos definen a una familia de sistemas en términos de un patrón de organización estructural, es decir, un estilo arquitectónico establece todos los componentes y conectores que se pueden establecer en determinadas instancias, así como las restricciones.

**Las estructuras que se desarrollan** a partir del proceso de arquitectura del software **deben ser desarrolladas de acuerdo con el tipo de sistema de software que se desea**, esto se conoce a través de los requisitos (Rani, 2017). Asimismo, (Rani, 2017) presenta 7 tipos de estilos arquitectónicos, que a su vez cada uno de esos contiene subtipos de arquitecturas.

- ✓ **Arquitectura de flujo de datos** - Esta arquitectura consiste en un método que sirve para determinar el cálculo paralelo en grano fino, en donde las direcciones accesibles para la ejecución simultánea se componen una muy cerca de la otra (Rani, 2017). Entre los subtipos de arquitectura de flujo de dato se encuentra: 1) Arquitectura secuencial por lotes y 2) Arquitectura de tuberías y filtros.
- ✓ **Arquitectura distribuida** - Esta arquitectura admite que diferentes sistemas de procesos funcionen como uno solo en diferentes unidades y lugares (Rani, 2017). Es decir, esta arquitectura admite que de forma coordinada se pueda tener acceso a la información desde cualquier parte del mundo. Los subtipos de arquitectura distribuida son: 1) Arquitectura cliente servidor, 2) Arquitectura Broker (CORBA, por sus siglas del inglés *Common Object Request Broker Architecture*) y 3) Arquitectura Orientada a Servicios (SOA, por sus siglas del inglés *Service Oriented Architecture*).
- ✓ **Arquitectura orientada a la interacción** - Esta arquitectura está orientada a involucrar las interacciones de entrada y salida de usuario con las interfaces de

usuario específicas para sistemas de software (K. Qian et al., 2009). Es decir, trata de aislar al cliente sobre la preparación de la información mediante tres componentes: módulo de datos, módulo de control y módulo de presentación (Rani, 2017). El subtipo de arquitectura que se orienta a la interacción es la: arquitectura Modelo-Vista-Controlador (MVC).

- ✓ **Arquitectura centrada en datos** - Esta arquitectura presenta un almacén de datos centralizado el cual es compartido por diferentes componentes de software, busca lograr una integración de la información (Rani, 2017). El módulo de datos de esta arquitectura ofrece diferentes mecanismos para que los componentes de software puedan acceder a él (inserción, eliminación, actualización y recuperación) (K. Qian et al., 2009). Entre los subtipos de arquitectura centrada en datos se encuentra: 1) Arquitectura de repositorio y 2) Arquitectura de pizarra.
- ✓ **Arquitectura jerárquica** - Esta arquitectura organiza todo el sistema de software como una estructura jerárquica, en donde el sistema se descompone en subsistemas con distintos niveles de jerarquía (Rani, 2017). Cada módulo de nivel superior invoca procedimientos a módulos de nivel inferior, el cual le provee los servicios solicitados (K. Qian et al., 2009). Entre los subtipos de arquitectura jerárquica se encuentran: 1) Arquitectura de llamada y retorno, 2) Arquitectura en capas y 3) Arquitectura de máquina virtual.
- ✓ **Arquitectura basada en componentes** - Esta arquitectura se caracteriza por dividir un problema en subproblemas, cada uno de ellos es asociado con particiones de componentes. Por tanto, esta arquitectura promueve la reutilización de componentes en diferentes instancias del mismo proyecto o de otros proyectos (Rani, 2017).
- ✓ **Arquitectura de comunicación implícita-asíncrona** - Esta arquitectura emplea comunicaciones implícitas asíncronas para que un evento invocado puede comunicarse en una o más ocasiones. De esta forma, desde que se

informa la ocasión, el sistema invoca todos los métodos que están inscritos para esa ocasión (K. Qian et al., 2009; Rani, 2017). Los subtipos de arquitectura de comunicación implícita-asíncrona son: 1) Arquitectura basada en mensajes y 2) Arquitectura basada en eventos.

#### 4.5.3. Diagramas de flujo de datos

Los diagramas de flujo de datos (DFD, por sus siglas del inglés *Data Flow Diagram*) consiste en una metodología que se puede aplicar para diseñar las dependencias funcionales de un sistema de información (Butler et al., 1995; Soudani et al., 2009).

Los DFD permiten rastrear los movimientos de los datos de entrada, salida y almacenamiento (Olayan et al., 2013). Lo anterior, permite una mayor comprensión de las acciones de los sistemas de información. Para (Alshamrani & Bahattab, 2010) los DFD deben cumplir con las siguientes restricciones:

- ✓ Un flujo de datos proveniente de (ir a) un almacén de datos o un terminador debe ir a (provenir) de un proceso.
- ✓ Un proceso debe tener al menos un flujo de datos entrantes y al menos un flujo de datos salientes.
- ✓ Un terminador de entrada debe tener al menos un flujo de datos salientes, pero ningún flujo de datos entrantes.
- ✓ Un terminador de salida debe tener al menos un flujo de datos entrantes, pero no flujos de datos salientes.
- ✓ Un nodo auxiliar tiene un flujo de datos entrantes y al menos dos flujos de datos salientes que representan las bifurcaciones en el flujo de datos entrantes.

También, los DFD pueden ser representados a través del Lenguaje Unificado de Modelado (UML, por sus siglas del inglés *Unified Modeling Language*) empleando diagramas de actividad (Sommerville, 2016).

#### 4.6. Codificación del software

La implementación del software se encarga de traducir todas las especificaciones diseñadas en código fuente. Que los sistemas de software por desarrollar sean fáciles de leer y comprender. La escritura del código fuente es fundamental, entre más fácil y menos líneas se utilicen simplifican los trabajos de pruebas y mantenimiento, por ende se debe considerar obtener un estilo de programación, pautas de codificación, técnicas de codificación estructuradas y técnicas de verificación del código (Prasad & Verma, 2016).

##### 4.6.1. Características del código del software

Prasad & Verma (2016) destacan que el código del software debe estar escrito en congruencia con los requisitos del usuario, el programa se va a destacar como bueno si el código es perfecto o contiene un mínimo margen de error. Para apoyar la codificación del software, estos autores (Prasad & Verma, 2016) presentan una serie de características que ayudaran en aumentar el rendimiento del software.

- ✓ **Simplicidad** - El código del software debe escribirse de manera simple y concisa, la simplicidad debe mantenerse en la organización, implementación y diseño del código del software.
- ✓ **Modularidad** - Dividir el software en varios módulos, hace que sea fácil de entender y también fácil de depurar. Con la función de modularidad, el mismo segmento de código puede reutilizarse en uno o más programas de software.
- ✓ **Diseño** - El código del software está diseñado correctamente si se presenta de manera adecuada. El diseño del software debe decidirse antes de comenzar a escribir el código del software. Escribir el código del software en un estilo específico y consistente ayuda a otros desarrolladores de software a revisarlo.

- ✓ **Eficiencia** - Se dice que un programa es eficiente si hace un uso óptimo de los recursos disponibles.
- ✓ **Claridad** - Los códigos de software deben ser claros para que los desarrolladores puedan comprender el programa sin ninguna complejidad. La claridad se puede lograr mediante el uso de características tales como simplicidad, legibilidad y modularidad. Tenga en cuenta que la claridad comprende la claridad del código, la claridad del diseño y la claridad del propósito para conocer lo que ocurre en cada nivel del programa de software.
- ✓ **Accesibilidad** - Los códigos de software deben escribirse de manera que los componentes de software (por ejemplo, archivos y funciones) estén fácilmente disponibles y accesibles. Para que los archivos y funciones sean accesibles, deben tener nombres significativos, así como subtítulos y texto de soporte para cada imagen. Del mismo modo, debe haber hipervínculos y ayudas de navegación para ayudar a los usuarios a buscar información de diferentes secciones del código del software.
- ✓ **Estabilidad** - Se dice que los códigos de software son estables si pueden funcionar correctamente en diferentes plataformas sin afectar su diseño y consistencia. Para verificar la estabilidad, los códigos de software deben ser probados por errores e inconsistencias.

#### 4.6.2. Reutilización de software

La reutilización de código de software ha existido durante décadas, sin embargo, en los últimos años ha existido una tendencia a reutilizar hasta generar enfoques para organizar código y poder ser incorporado en diferentes sistemas, por ejemplo, bibliotecas de módulos, fragmentos de código o clases (Ralph & Wand, 2007).

Por consiguiente, la reutilización de software es ahora mismo una norma para muchas empresas y organizaciones para el desarrollo de diferentes tipos de

sistemas de software, principalmente para sistemas basados en la web. Además, en la Figura 5.3, se puede observar que la reutilización del software es posible de implementar en diferentes niveles: Nivel de abstracción, nivel del objeto, nivel del componente y nivel del sistema (Sommerville, 2016).



**Figura 5.3 Reutilización de software**

**Fuente:** Elaboración propia a partir del texto de (Sommerville, 2016)

- ✓ **Nivel de abstracción** - A este nivel no se reutiliza software, se utiliza el conocimiento de abstracciones exitosas para el diseño de software a través de los patrones de diseño y patrones arquitectónicos.
- ✓ **Nivel del objeto** - En este nivel se reutilizan objetos de una biblioteca, por ejemplo, objetos y métodos de una biblioteca de JavaMail.
- ✓ **Nivel del componente** - A este nivel se entiende por componentes como colecciones de objetos y clases de objetos que funcionan en conjunto para proporcionar funciones y servicios.
- ✓ **Nivel del sistema** - En este nivel se reutilizan sistemas completos, por lo general se realizan reconfiguraciones, como agregando o modificando el código.

## 4.7. Pruebas de software

Las pruebas de software consisten en una estrategia incluida como parte de la gestión de riesgos, en donde se verifica que todos los requisitos funcionales obtenidos hayan sido cumplidos, entre las principales actividades que se incluyen en esta estrategia se busca crear una verificación y validación del software (Lewis, 2017). Las pruebas de software van más allá de las estrategias planteadas para el diseño y codificación del sistema de software, es parte de una política del proyecto e institución garantizar la calidad del producto.

Las pruebas de software son críticas, debido a que determinan la corrección, integridad y calidad del software que se está desarrollando. Se debe plantear una estrategia que proporcione las condiciones y la ruta correcta para evaluar el software. Por consiguiente, los investigadores de Oliveira & Senger de Souza (2019) y Prasad & Verma (2016) concuerdan en que cualquier estrategia de software debe incorporar al menos 4 pasos:

- 1) **Planificación**, en este paso se crea el plan de las pruebas,
- 2) **Diseño** del caso de prueba, en este paso se establecen los casos de prueba,
- 3) **Ejecución**, este paso ejecuta los casos de prueba planificados y se registran los resultados, y
- 4) **Análisis**, este paso analiza los resultados de las pruebas para garantizar que se cumple con las especificaciones.

### 4.7.1. Técnicas de pruebas de software

Las prueba de software han estado activas durante las últimas dos décadas, y conforme el tiempo avanza los fundamentos y requerimientos van cambiando, con ello las formas de evaluar software ha evolucionado (Ammann & Offutt, 2008). Sin embargo, es importante exponer las principales terminologías empleadas durante los años para evaluar software, como lo son las pruebas de caja negra, caja blanca y caja gris, aunque actualmente se encuentra obsoleta.

- ✓ **Pruebas de caja negra** - Las pruebas de caja negra hacen referencia a lo funcional, son pruebas que buscan verificar la funcionalidad de un programa o

un sistema de software, por tanto, el encargado de probar se encarga únicamente de validar la información con datos de entrada y salida, y no como funciona internamente el programa (Lewis, 2017). El nombre de pruebas de caja negra refiere porque el probador ve el programa como una caja negra, no se preocupa del comportamiento interno, ni estructura, únicamente se enfoca en la funcionalidad del programa contra las especificaciones obtenidas de acuerdo con los requisitos recolectados (Myers et al., 2012).

- ✓ **Pruebas de caja blanca** - Las pruebas de caja blanca son las pruebas que permiten evaluar de forma estructural a un programa o sistema de software. Este tipo de pruebas se realizan a través de una ruta lógica, se examina la lógica del programa sin necesidad de preocuparse o considerar a fondo los requerimientos (Myers et al., 2012). Las pruebas de caja blanca son exhaustivas y están centradas en el código (Lewis, 2017).
- ✓ **Pruebas de caja gris** - Este tipo de pruebas consiste en la combinación de pruebas funcionales y estructurales. Se da en menor escala, sin embargo, en ciertas situaciones el probador podría llegar a identificar y comprender no solo las especiaciones, sino también la estructura interna del programa y con ello, poder diseñar pruebas implícitas (Lewis, 2017).

#### 4.8. Conclusiones del capítulo

En este capítulo se abordaron consideraciones teóricas que se deben tomar en cuenta al desarrollar un software determinado. Entre las consideraciones más relevantes están:

Se evidencia que, al diseñar un sistema informático, los programadores deben de determinar 3 tareas, vinculadas con definir claramente las instrucciones por realizar, el orden y los datos requeridos para ejecutar las instrucciones.

En relación con las categorías de programación (no estructurada, estructurada y orientada a objetos), destacándose la programación orientada a objetos por permitir la reutilización de piezas de software, la cual se puede realizar desde

distintos niveles: Nivel de abstracción, nivel del objeto, nivel del componente y nivel del sistema.

En cuanto a los modelos de desarrollo de software, el DCU se destaca por enfocarse en las necesidades, limitaciones y los requerimientos directamente de los usuarios finales.

# **CAPÍTULO VI. REVISIÓN DE METODOLOGÍAS UTILIZADAS PARA EL DISEÑO DE JUEGOS SERIOS**

#### 4.9. Introducción

Este capítulo presenta y describe algunas de las principales metodologías que han servido de guía para el diseño y desarrollo de videojuegos y, en particular, juegos serios. Metodologías identificadas de acuerdo con la revisión de literatura existente y, desde el punto de vista de diferentes autores (Cano et al., 2016; Spinelli et al., 2017; Spinelli & Massa, 2018b, 2018a; Tran et al., 2010), entre otros.

Luego, se definen un conjunto de criterios que viabilizarán la indagación de acuerdo con los objetivos de este trabajo, los cuales se han agrupado en 4 categorías de análisis (aspectos generales, aspectos de diseño, aspectos metodológico-pedagógicos y aspectos de análisis). Se incluye una descripción de cada metodología seleccionada, en la que se presenta el objetivo y sus principales aspectos a nivel de diseño.

Seguido, se ofrece un resumen de dichos aspectos identificados en cada metodología acorde a los resultados del análisis y bajo la luz de los criterios definidos. Luego, se realiza un análisis global de las metodologías descritas acorde a cada categoría de análisis establecida.

Finalmente, se plasman conclusiones que permiten resaltar una serie de aspectos importantes que han sido considerados dentro de las diferentes propuestas metodológicas analizadas.

Los contenidos del capítulo permiten dar respuesta a las preguntas de investigación; Plx10 y Plx11 establecidas para el estudio y referidas en el Capítulo 1:

Plx10 ¿Qué metodologías se han utilizado para el diseño de juegos serios?, ¿qué antecedentes existen en este sentido?

Plx11 ¿Qué aspectos metodológicos se consideran en la literatura para el desarrollo de juegos serios?

En la próxima sección se definen y describen los criterios que servirán de guía para realizar el análisis de las metodologías seleccionadas para el estudio.

#### 4.10. Definición de los criterios de análisis

A través de la revisión de literatura, se evidencia la utilización de diferentes tipos de metodologías/procesos para el diseño de juegos serios, mismos que varían

de acuerdo con el autor o investigador que las plantea y del objetivo propio de cada investigación. Con este fin, se determinan criterios para orientar el análisis, de manera homogénea, de un grupo de casos. La selección de los criterios se basa en (Cano, 2016; Cano et al., 2016) y en el objetivo de estudio que busca describir metodologías que han sido utilizadas para el diseño de juegos serios, con especial atención en los aspectos que se deben considerar para su diseño.

Para describir y analizar las metodologías o procesos, se definen 4 categorías con diferentes criterios de análisis cada una:

- A. Aspectos generales** - Los criterios de esta categoría son afines a las metodologías por analizar. Los indicadores, permiten identificar el país de procedencia, el nivel educativo.
- B. Aspectos de diseño** - Los criterios de esta categoría están vinculados a analizar la forma en que se definen aspectos del diseño de las metodologías/procesos. Se busca identificar si se realiza unalicitación y definición de requerimientos de los usuarios, conocer si las metodologías dentro de sus propuestas incluyen la definición de roles, así como la definición de patrones de diseño, reutilización de software, documentación y diseño de prototipos.
- C. Aspectos metodológico-pedagógicos** - Los criterios presentes en esta categoría permiten conocer los destinatarios, la intención pedagógica y la definición de los objetivos pedagógicos/lúdicos de la metodología.
- D. Aspectos de análisis** - Esta categoría incluye los criterios que analizan la actividad de desarrollo de la metodología. Por tratarse de propuesta metodológicas o de procesos para el diseño de juegos serios educativos, se busca conocer el proceso o método utilizado para la validación de los objetivos de la propuesta, así como la evaluación de las experiencias del usuario, y, finalmente, la utilización de resultados cuantificables que evalúen la calidad del aprendizaje del usuario.

En la [Tabla 6.1](#), se puede observar la clasificación de los criterios en cada una de las categorías detalladas (aspectos generales, aspectos de diseño, aspectos metodológico-pedagógicos y aspectos de análisis).

**Tabla 6.1** Criterios de análisis de las metodologías para el diseño de juego serios

Categoría	Criterios
Aspectos generales	País donde desarrolla la investigación. Nivel educativo.
Aspectos de diseño	Fundamentación. Definición de requerimientos. Definición de roles. Definición de patrones de diseño. Reutilización de software. Documentación. Definición de prototipos.
Aspectos metodológico-pedagógicos	Definición de destinatarios de la metodología. Intención pedagógica. Definición de objetivos pedagógicos/lúdicos.
Aspectos de análisis	Validación de objetivos. Evaluación de las experiencias del usuario. Resultados cuantificables.

Seguido, se realiza una descripción de los criterios de análisis propuestos para la revisión y estudio de las metodologías con el propósito de brindarle semántica a cada uno y posteriormente, hacer más fácil su comprensión.

#### A. Aspectos generales

En esta categoría se consideran dos criterios: País donde desarrolla la investigación y el nivel educativo:

- ✓ **País donde desarrolla la investigación (origen de la propuesta)** - Este criterio hace referencia al país donde se desarrolla la metodología.
- ✓ **Nivel educativo** - Este criterio permite identificar el nivel educativo al que está dirigida la metodología. Puede ser a nivel de educación especial, primaria, secundaria o superior/universitaria (grado y postgrado).

#### B. Aspectos de diseño

La categoría de aspectos de diseño reúne 7 criterios: Fundamentación, definición de requerimientos, definición de roles, definición de patrones de diseño, reutilización de software, documentación y definición de prototipos:

- ✓ **Fundamentación** - Este criterio permite identificar el fundamento o base que los investigadores han tomado de referencia para el diseño de la

metodología. Los posibles valores para este criterio son; DCU, ADDIE, MPOBA, entre otros.

- ✓ **Definición de requerimientos** - Este criterio permite identificar si la propuesta metodológica considera la elicitación y definición de requerimientos por parte del usuario. Los posibles valores para este criterio son; **sí / no**. Técnica utilizada (identifica técnicas de levantamiento y especificación de requisitos empleada).
- ✓ **Definición de roles** – Este criterio permite identificar los roles del equipo multidisciplinario que han sido propuestos en cada metodología/procedimiento. Los valores para este criterio podrían ser: docentes, expertos en contenidos, psicólogos, estudiantes, diseñadores, desarrolladores, entre otros.
- ✓ **Definición de patrones de diseño** - Este criterio permite identificar si la propuesta metodológica/procedimiento incluye patrones de diseño, los cuales son de utilidad para establecer un lenguaje de comunicación entre los diferentes actores que se encuentran involucrados con el desarrollo del juego serio. Los posibles valores para este criterio son; **sí / no**.
- ✓ **Reutilización de software** - Este criterio permite identificar si en las metodologías analizadas se considera la reutilización de software. Los posibles valores para este criterio son; **sí / no**.
- ✓ **Documentación** – Este criterio permite identificar si en las metodologías analizadas se contempla la documentación de todo el proceso. Los posibles valores para este criterio son; **sí / no**. Tipo de documentación utilizada (documentación de análisis, diseño, producción, entre otros).
- ✓ **Prototipos** - Este criterio permite identificar si las propuestas metodológicas incluyen el diseño de prototipos de software, los cuales sirven para identificar problemas de usabilidad y evaluar el producto, a través de la evaluación del alcance de los objetivos pedagógicos y lúdicos. Estos prototipos permiten realizar evaluaciones por parte de expertos en el dominio y usuarios finales. Los posibles valores para este criterio son; **sí / no**.

## C. Aspectos metodológico-pedagógicos

Esta categoría engloba 3 criterios: Definición de destinatarios de la metodología, intención pedagógica y definición de objetivos pedagógicos/lúdicos:

- ✓ **Destinatarios de la metodología** - Este criterio hace referencia al público meta para el cual está orientada la metodología, es decir, la población en específico, tal como desarrolladores, programadores, docentes o estudiantes.
- ✓ **Intención pedagógica** - Este criterio hace referencia al objetivo principal que caracteriza la metodología, es decir, la habilidad o competencia que se pretende potenciar. Los valores posibles por obtener se vinculan con la formación o adquisición de habilidades o competencias (competencias genéricas, digitales, ambientales, salud, educación, entre otras) que se desean alcanzar en algún área en específico (área del conocimiento para las cuales han sido diseñadas).
- ✓ **Definición de objetivos pedagógicos/lúdicos** – Este criterio permite determinar si se han propuesto y definido objetivos pedagógicos/lúdicos por alcanzar con los estudiantes y conducentes al logro de la intención pedagógica. Los posibles valores para este criterio son; **sí / no**. Tipo de objetivo (competencias, conocimiento, entrenamiento o cambios de actitud, entre otros).

#### D. Aspectos de análisis

En esta categoría se incluyen 3 criterios: Validación de objetivos, evaluación de las experiencias del usuario y los resultados cuantificables:

- ✓ **Validación de los objetivos** – Este criterio permite determinar si la metodología considera realizar una evaluación de aspectos vinculados con la producción del juego y el cumplimiento de los objetivos. Los posibles valores para este criterio son; **sí / no**. Tipo de herramienta utilizada para tal propósito.
- ✓ **Evaluación de las experiencias del usuario** - Este criterio permite determinar si la metodología realiza una evaluación a nivel de las experiencias del usuario, mediante la cual se identifican problemas y

aspectos de diseño, usuario y juego. Los posibles valores para este criterio son; **sí / no**. Tipo de objetivo.

- ✓ **Resultados cuantificables** - Este criterio permite analizar si en las experiencias seleccionadas se identifica que hayan sido sometidas a un proceso de evaluación del aprendizaje y la experiencia del usuario. Los posibles valores para este criterio son; **sí / no**. Tipo de elemento pedagógico utilizado.

#### 4.11. Recopilación de las metodologías

A partir de la revisión de literatura, se identifica un conjunto de metodologías/procesos para el diseño de juegos serios (MDJS), las cuales han sido seleccionadas para el análisis y se pueden observar en la [Tabla 6.2](#).

**Tabla 6.2 Metodologías seleccionadas para análisis**

a Referenci a	Metodología	a Referenci a	Metodología
MDJS1	EMERGO	MDJS6	VGSCL
MDJS2	EDoS	MDJS7	MECONESIS
MDJS3	LEGADEE	MDJS8	MPIu+a
MDJS4	SAVIE	MDJS9	MPDSG
MDJS5	DODDEL		

A continuación, se describe brevemente cada una de las metodologías citadas, las cuales han sido diseñadas en diferentes países.

##### 4.11.1. Metodología EMERGO

En los **Países Bajos** (Nadolski et al., 2008), investigadores asociados a la Universidad Abierta de los Países Bajos, desarrollan una metodología compuesta por diferentes fases y un conjunto de herramientas cuyo propósito es servir de guía para el diseño de juegos serios basados en escenarios. Metodología a la cual le han asignado el nombre de **EMERGO**, la cual está pensada para ser utilizada principalmente por docentes para potenciar en el estudiantado competencias más de tipo genérico, relacionadas con la toma de decisiones, resolución de problemas, pensamiento crítico, entre otras habilidades cognitivas complejas.

La metodología EMERGO tiene como propósito guiar el desarrollo de juegos serios basados en escenarios. Esta metodología toma como base el enfoque por

fases para materiales de instrucción, conocido como ADDIE, acrónimo de *Analysis, Design, Development, Implementation and Evaluation* (análisis, diseño, desarrollo, implementación y evaluación) (Plomp et al., 1992), el cual fue adaptado al proceso de diseño de juegos serios (Padilla-Zea, 2011).

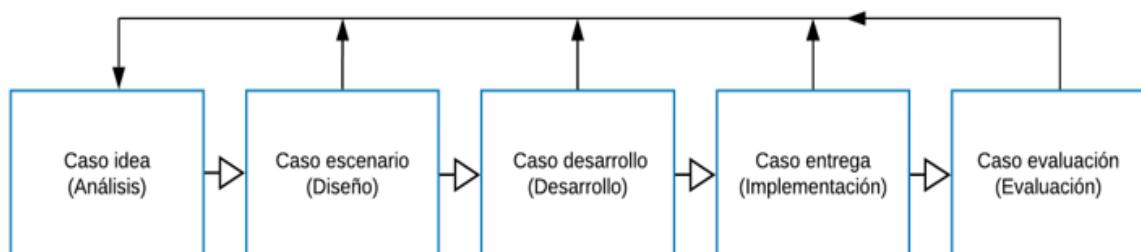
Estos autores (Nadolski et al., 2008) definen los juegos serios basados en escenarios como un entorno de tareas simuladas vinculadas con situaciones de la vida real, donde se desarrollan secuencias de actividades de aprendizaje que implican procesos de toma de decisiones, definir estrategias para la resolución de problemas, utilizar el razonamiento inteligente, entre otras habilidades cognitivas (Evans et al., 2016; Massa et al., 2017).

Si bien, las fases propuestas en EMERGO se podrían ejecutar en el orden previamente establecido, es recomendable la utilización de un proceso iterativo. Además, las herramientas ofrecidas por esta metodología se utilizan durante las fases de desarrollo e implementación.

En EMERGO, la metodología y el conjunto de herramientas ofrecidas, están destinadas para ser utilizadas principalmente por el profesorado, quienes podrían ser acompañados o asistidos por especialistas de otras áreas del conocimiento, tales como tecnólogos educativos, diseñadores gráficos o de juegos, quienes podrían servir de apoyo durante el desarrollo de las primeras 4 fases (pasar de la idea del caso a la entrega del caso).

En relación a la usabilidad de la metodología y sus herramientas, van a depender de diferentes factores o circunstancias; la complejidad del caso, el contenido a desarrollar o enseñar y los procedimientos de trabajo de la organización (Massa et al., 2017; Nadolski et al., 2008; Padilla-Zea, 2011).

En la [Figura 6.1](#), se puede observar gráficamente las fases de diseño y desarrollo propuestas en EMERGO.



**Figura 6.1 Metodología para el desarrollo de casos EMERGO**

**Nota:** Los triángulos abiertos indican un orden sugerido, pero las fases se pueden realizar de forma iterativa (flechas negras). **Fuente:** Elaboración propia a partir del texto de (Nadolski et al., 2008)

A continuación, se citan y describen a nivel general las fases propuestas en la metodología EMERGO (Nadolski et al., 2008; Padilla-Zea, 2011):

1) **Análisis** – En esta fase, se debe resolver por parte de los desarrolladores de casos (juegos), algunos problemas previos vinculados al caso inicial. Para ello, se recomienda un debate por parte del grupo de trabajo, mediante el cual se pueda extraer más información, encontrar puntos en común y generar una concientización en términos de la importancia del caso, los objetivos de aprendizaje de este y la forma en la que será estructurado.

Es decir, debe existir entre el grupo de desarrolladores una visión clara de los objetivos de aprendizaje y alcances del proyecto, entender el ¿qué?, ¿por qué? y el ¿cómo? Antes del inicio del diseño y desarrollo del caso, se debe poseer un panorama real de las posibilidades y limitaciones de este.

Luego, con la intención de proporcionar una descripción global del caso, la cual servirá como documento de entrada para la fase de diseño, se recomienda brindar respuesta a una serie o subconjunto de preguntas, mismas que se pueden observar en la [Tabla 6.3](#).

2) **Diseño** – En esta fase, se debe generar como resultado un documento de **escenario detallado** a través de la realización de los pasos intermedios de **marco de trabajo de escenario e ingredientes de escenario**, estos pasos aportan mayor detalle e integridad (completitud). A continuación, se describe brevemente los 3 escenarios:

✓ **El marco de trabajo de escenario** - Se realiza una descripción general de las actividades desarrolladas por parte del estudiantado durante el juego. Estas actividades son anotadas mediante la utilización del formato estándar conocido como *Where the student will* (descripción de la actividad), en la cual se han identificado tres ventajas de utilizarlo: *i.* Permite la identificación de un conjunto de actividades sin distraerse con los detalles, lo cual favorece la identificación de actividades obligatorias y no obligatorias, ordenadas y no ordenadas,

esperadas e inesperadas. *ii.* El escenario es similar a un plano de una construcción, en el cual se muestran los aspectos por trabajar más adelante. *iii.* Brinda el mismo nivel de atención o importancia a todos los elementos durante las diferentes etapas de diseño y desarrollo. El escenario marco es flexible, lo cual permite realizar ajustes durante los siguientes pasos.

- ✓ **Los ingredientes de escenario** - Se realiza para todas las etapas de la actividad, una identificación de las posibles actuaciones o desempeño por parte de los jugadores (estudiantes), es decir, en este paso es importante brindar respuesta a diferentes interrogantes: ¿Qué hacen los estudiantes, con quién interactúan, con qué herramientas y recursos, con qué apoyo (profesores, compañeros, otros)? ¿Las tareas resultan en un producto? Si es así, ¿cómo se evaluará este producto? ¿Se necesita un resultado positivo o satisfactorio antes de que los estudiantes puedan continuar? ¿Qué interacciones con otros participantes y el programa se prevén durante y después de realizar actividades? Finalmente, se debe describir exhaustivamente todas las interacciones para cada actividad, se debe tener en cuenta que, esta descripción todavía no se debe realizar en términos de herramientas y recursos.
- ✓ **El escenario detallado** - Se debe realizar una descripción de cada actividad de forma exhaustiva, en términos de herramientas y recursos que se necesitan para su completo desarrollo. Es decir, si la población estudiantil requiere realizar entrevistas, todas las preguntas deben estar identificadas. Igualmente, si los estudiantes requieren de algún tipo de recurso, todos esos recursos deben estar identificados. En este paso, es de vital importancia tener claro y conocer si los materiales del caso ya están disponibles o bien, se deben desarrollar. Es importante considerar para el desarrollo de estas actividades, la utilización de diferentes componentes y herramientas EMERGO. Además, para el desarrollo de las actividades de este paso, se recomienda utilizar el software de diagramas de flujo y programación.

- 3) **Desarrollo** – En esta fase, se utilizan las herramientas de EMERGO para introducir los datos. Para tal propósito, se puede hacer uso del escenario detallado como guía u orientación. En este caso, la totalidad de los componentes de entrada de datos de las herramientas EMERGO proporcionan plantillas para guiar dicho proceso. Para la introducción de datos no se requiere de una experticia específica, sin embargo, no se puede prescindir por completo.
- 4) Además, se requiere de la realización de pruebas sistemáticas, usualmente, algunos de los problemas que se identifican, solo pueden ser resueltos por expertos en dominios o expertos en secuencias de comandos más complejas. Por consiguiente, se debe realizar un trabajo conjunto y consensuado entre los equipos de diseño y desarrollo.

Los tres roles que se distinguen durante el desarrollo son:

- ✓ **Propietario del caso (juego)** - Responsable de definir cuáles serán los derechos de acceso de cada componente del juego.
- ✓ **Autor del componente** - Responsable de la entrada de datos de una parte específica del juego.
- ✓ **Probador del juego** - Requiere de poder cambiar de rol en el juego para probar distintas características.

El estudiantado requiere compartir, debatir y almacenar sus experiencias o vivencias con el juego, para el desarrollo de un aprendizaje efectivo. Para el registro de los resultados de las experiencias, el estudiantado puede tomar notas a través de:

- ✓ **Colocar notas en ubicaciones (mapa)** - Esta tarea se podría realizar durante el desarrollo de una conversación o al momento de leer un documento.

- ✓ **Organizar y procesar las notas tomadas** - De forma automática se clasifican y almacenan en bitácoras las notas que han sido tomadas por cada estudiante. Posteriormente, el estudiante puede utilizar estas bitácoras para resumir las notas y generar informes.
- ✓ **Informes finales (reportes)** - Se podrían elaborar informes como productos de tareas ejecutadas a través del libro de registros o portapapeles. Estos informes podrían ser cargados mediante e-Mensajes (eMessages).
- ✓ **Construir un eMessage** - Se requiere únicamente adjuntar un informe.

Con relación a la **prueba de entrada de datos**, esta implica realizar intercambios entre personajes del juego. Esta prueba se puede realizar de tres formas:

- ✓ Iniciar y usar los resultados de la sesión anterior (predeterminado).
- ✓ Iniciar y no usar los resultados de otra sesión.
- ✓ Iniciar y usar los resultados de otra sesión (no la anterior) (usando ejecuciones de demostración).

Tabla 6.3 Preguntas para la fase de análisis

Categoría	Preguntas
Contexto	<p>¿Para qué curso, currículo o instituciones se va a usar?</p> <p>¿Es un material aislado o se usará con otros materiales?</p> <p>¿Qué carga de estudio e intervalo de tiempo es el esperado?</p> <p>¿Cuánta recompensa gana el estudiante cuando lo supera?</p> <p>¿Cuál es la principal habilidad cognitiva que se requiere?</p> <p>¿Se necesitan otras habilidades?</p> <p>¿Qué conocimiento previo y habilidades necesitan los alumnos?</p> <p>¿Qué es lo fundamental (por ejemplo, el paciente, el equipo, entre otros)?</p> <p>¿Cuáles son las localizaciones (por ejemplo, mapear espacios virtuales)?</p> <p>¿Qué características son relevantes?</p> <p>¿Los estudiantes necesitan seguir un proceso establecido para realizarlo?</p> <p>¿Qué tipo de actividades necesitan realizar los estudiantes para adquirir las habilidades cognitivas esperadas?</p> <p>¿Hay un orden estricto en las tareas obligatorias?</p> <p>¿Hay tareas obligatorias y voluntarias y, qué determina que sean de un tipo u otro?</p>
Contenido	<p>¿La información suministrada es estrictamente necesaria o hay redundancias?</p> <p>¿Cómo de realista y auténtico es el caso?</p> <p>¿Si los estudiantes pueden repetir un caso, será el mismo caso o habrá variaciones?</p> <p>¿Los estudiantes pueden deshacer decisiones previas?</p> <p>¿Hay diferentes rutas de aprendizaje y tareas para diferentes estudiantes?</p> <p>¿Qué tipo de cooperación necesitan los estudiantes?</p> <p>¿Los estudiantes tienen diferentes características en sus casos?</p> <p>¿Los estudiantes tienen roles activos?</p> <p>¿Los profesores tienen roles activos?</p> <p>¿Se ha incorporado la competencia? ¿Cómo obtienen los estudiantes recompensas por excelente desempeño o comportamiento?</p> <p>¿Cómo descubren los estudiantes que todavía no han adquirido las habilidades cognitivas esperadas?</p> <p>¿Cómo pueden los estudiantes monitorizar su progreso?</p> <p>¿Cómo se comprueba si los estudiantes han adquirido las habilidades cognitivas esperadas?</p> <p>¿Se ha incluido evaluación sumativa? ¿Se usa el resultado para una evaluación formativa?</p> <p>¿Qué cifras de progreso de los alumnos se van a usar para informar a los profesores durante la ejecución del juego?</p>
Progreso del estudiante	<p>¿Se debe promover el contacto entre compañeros?</p> <p>¿Deben los alumnos poder ver si sus compañeros están conectados?</p> <p>¿Los estudiantes pueden comparar su progreso con los compañeros?</p> <p>Si se ha usado el material, ¿se necesita material nuevo?</p> <p>¿Qué recursos se utilizan (entrevistas, películas, animaciones...)?</p> <p>¿Qué medios se necesitan y cuáles son sus costos?</p>
Contacto con compañeros	<p>¿Está restringido el número de estudiantes en cada ejecución?</p> <p>¿Cuándo puede un estudiante conectarse al juego?</p> <p>¿Es posible cambiar de caso una vez que se ha iniciado?</p> <p>¿Cómo se proporcionará el soporte técnico?</p>
Uso de medios	<p>¿Cómo se proporcionará el soporte para la adquisición de habilidades cognitivas?</p>
Entrega	<p>¿Cuántos estudiantes participarán cada año?</p> <p>¿Cuál es el costo de desarrollo por estudiante?</p> <p>¿Cuál es la proporción esperada de profesores / estudiantes durante la explotación?</p>
Soporte	
Costos	

Categoría	Preguntas
Derechos de propiedad intelectual	¿Está permitido que otros utilicen el caso? ¿Hay materiales incorporados por otros autores y cuáles son sus derechos sobre estos materiales?

Fuente: Tabla adaptada y traducida del inglés al español del texto de (Nadolski et al., 2008)

En la [Figura 6.2](#) y la [Tabla 6.4](#), se puede observar que los componentes para la entrada de datos se pueden usar tanto para la inicialización como para el desarrollo. La [Tabla 6.4](#), presenta un resumen de varios componentes de entrada de datos y su funcionalidad, los cuales se pueden apreciar de forma gráfica en la [Figura 6.2](#).

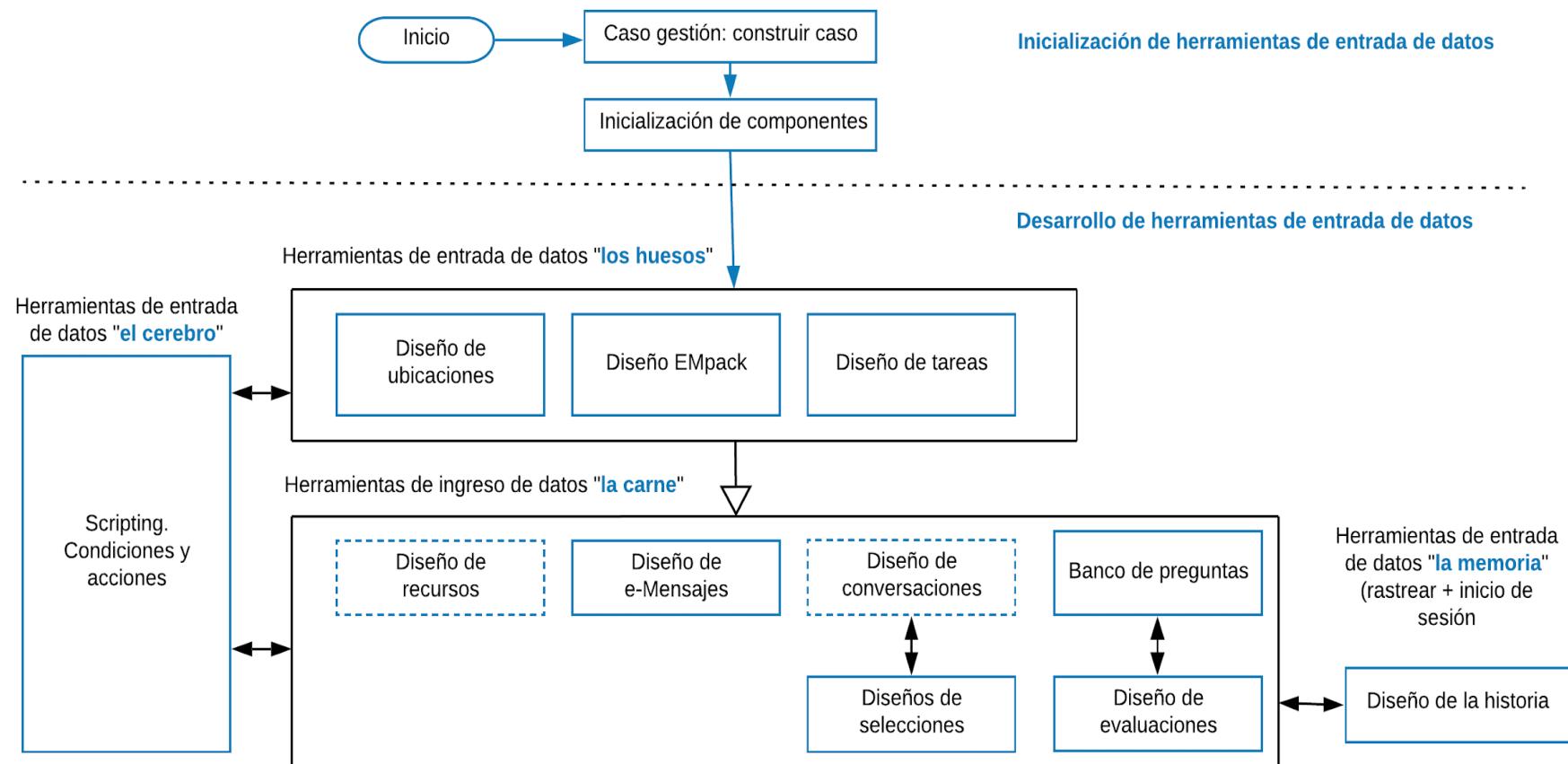
**Tabla 6.4 Componentes de entrada de datos y su función componente**

Componente	Función
Inicialización	
Gestión de caso: construir caso	Define nuevos casos. Los casos existentes se pueden usar para modificar.
Inicialización de componentes	Define los componentes y sus derechos de acceso para la entrada de datos.
Desarrollo	
Diseño de ubicaciones (locaciones)	Define ubicaciones y, posiblemente condiciones y acciones.
Diseño EMpack	Define herramientas en el EMpack y posiblemente, condiciones y acciones.
Diseño de tareas	Define tareas en la "lista de tareas" y posiblemente, condiciones y acciones.
Scripting	Define el flujo de casos por condiciones y acciones que se pueden especificar para el caso completo, uno o más componentes, o la entrada de datos específicos para uno o más componentes. Las condiciones y acciones a menudo se relacionan con las acciones de los estudiantes, pero también con las acciones de otros actores.
Diseño de recursos	Define los recursos y sus jerarquías.
Diseño de e-Messages	Define varios tipos de e-Mensajes.
Diseño de conversaciones	Definir conversaciones entre estudiantes y personajes virtuales.
Diseños de selección	Define opciones de selección para los jugadores cuando realizan una tarea.
Banco de preguntas	Define elementos de opción múltiple para evaluaciones (exámenes).
Diseño de evaluaciones	Define evaluaciones (exámenes).
Diseño de historias	Define las opciones para los registros de los estudiantes y como los estudiantes puedan recuperarlos.

Fuente: Tabla adaptada y traducida del inglés al español del texto de (Nadolski et al., 2008)

En el entorno del estudiante se pueden instanciar (usar) varias veces la mayoría de los componentes. Tales como, los **recursos de diseño** los cuales

permiten crear un **archivo de video**, un **archivador** o una **colección de recursos**.

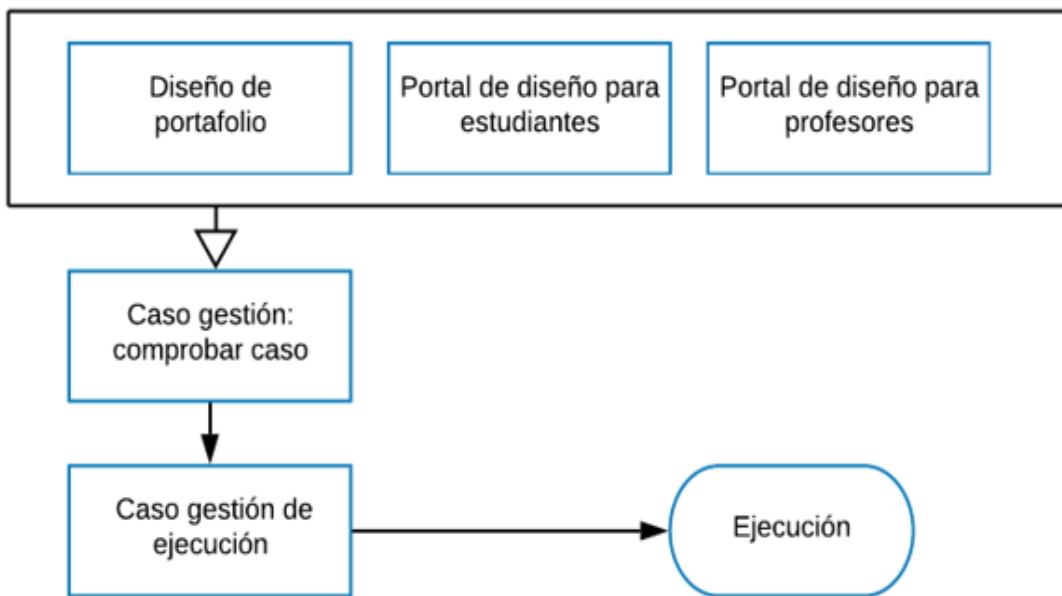
**Figura 6.2 Componentes de entrada de datos para el desarrollo de casos**

**Nota:** Las flechas con triángulos negros indican un orden obligatorio, la flecha con el triángulo abierto indica un orden sugerido. Para los componentes que están en un mismo cuadro, no existe orden sugerido. Las flechas bidireccionales indican relaciones entre componentes. El componente de secuencias de comandos está en el centro de toda la entrada de datos. Define el flujo de casos utilizando condiciones y acciones. **Fuente:** Elaboración propia a partir del texto de (Nadolski et al., 2008)

5) **Implementación (fase de entrega)** – En esta fase, se tiene acceso al juego por parte del estudiantado y profesorado. En esta etapa de entrega se supone que:

- ✓ Un estudiante puede elegir el caso (juego) en su entorno de estudio (portal del estudiante).
- ✓ Un profesor puede elegir el caso en su entorno de trabajo (portal del profesor).
- ✓ La entrada de datos para el caso ha sido verificada y es aceptable.
- ✓ La gestión de ejecución de casos se ha utilizado para preparar la ejecución del juego que se lanzará.

La [Figura 6.3](#), muestra los componentes de entrada de datos en las herramientas EMERGO para la entrega, mientras que la [Tabla 6.5](#) resume la entrada de datos con su funcionalidad. El juego podría ser publicado al cumplir con los supuestos de la etapa de entrega. Con los datos de autorización, tanto los estudiantes como los profesores pueden descargar los juegos desde la página web de EMERGO.

**Figura 6.3 Componentes de entrada de datos para entrega de casos**

**Nota:** Las flechas con triángulos negros indican un orden obligatorio, la flecha con el triángulo abierto indica un orden sugerido. Para los componentes que están en un mismo cuadro, no existe orden sugerido. **Fuente:** Elaboración propia a partir del texto de (Nadolski et al., 2008)

El juego podría ser publicado al cumplir con los supuestos de la etapa de entrega. Con los datos de autorización, tanto los estudiantes como los profesores pueden descargar los juegos desde la página web de EMERGO.

**Tabla 6.5 Componentes de entrada de datos y su función**

Componente	Función
Diseño de portafolio	Define qué productos para estudiantes se agregan al portafolio
Portal de diseño para estudiantes	Define qué casos se pueden elegir, qué datos de progreso de casos se pueden monitorear, comparaciones de progreso, entre otros.
Portal de diseño para profesores	Define qué casos se pueden elegir, qué datos de progreso del caso se pueden monitorear y para qué estudiantes, entre otros.
Caso gestión: comprobar caso	Comprueba si la entrada de datos es sintácticamente correcta.
Caso gestión de ejecución	Define ejecuciones de casos. Cada ejecución tiene una hora de inicio, estudiantes y profesores inscritos y, podría tener una hora de finalización.

**Fuente:** Tabla adaptada y traducida del inglés al español del texto de (Nadolski et al., 2008)

6) **Evaluación** – En esta fase, se mide y evalúa en los juegos el nivel de cumplimiento de los requerimientos explicitados a detalle en la fase de análisis, es decir, si el uso real obtenido del juego está acorde al uso esperado. En esta fase se deben contrastar las interrogantes planteadas en la fase de análisis

(Tabla 6.3). La Tabla 6.6, presenta un resumen acorde a los aspectos de análisis de EMERGO.

**Tabla 6.6 Resumen - Aspectos de análisis de EMERGO**

Aspectos generales	País Nivel educativo	Países bajos Superior/universitario
Aspectos de diseño	Fundamentación Requerimientos Definición de roles Patrones de diseño Reutilización de Software Documentación Diseño de prototipos	ADDIE No Propietario del juego, autor del componente, probador del JS No No A través de escenarios detallados No
Aspectos metodológico-pedagógicos	Destinatarios Intención pedagógica Objetivos pedagógicos/lúdicos	Docentes (principalmente) y estudiantes Adquisición de habilidades cognitivas Sí
Aspectos de análisis	Validación de objetivos Evaluación de UX Resultados cuantificables	Sí, Pruebas de entrada de datos, probador del juego y aplicación de cuestionario (preguntas en relación con el contenido y progreso del estudiante). No No

En suma, la metodología de diseño EMERGO enfatiza la interacción docente – estudiante mediante la cual se guía el diseño de juegos serios basados en escenarios orientados a la adquisición de habilidades cognitivas complejas en la a nivel de educación superior, donde los contenidos del juego son ejecutados como parte de una programación de actividades docentes que apoyan el desarrollo de procesos formativos mediado por juegos serios.

#### 4.11.2. Metodología EDoS

En **Francia** (Tran et al., 2010), investigadores de la Universidad de Lyon, proponen un entorno para el diseño de juegos serios llamado **EDoS** (*Environment for the Design of Serious Games*), donde las fases del proceso EDoS brindan una serie de herramientas interactivas para facilitar a los usuarios la realización de tareas de una manera más visual y fácil. El entorno de diseño y desarrollo de juegos serios EDoS, está vinculado a potenciar competencias relacionadas con la ingeniería.

EDoS, es un entorno que se utiliza principalmente en la fase de diseño y una parte de la fase de producción de juegos serios, es decir, se utiliza posterior a la fase de levantamiento de requerimientos de usuario y brinda como producto final un escenario estructurado y formal para ser ejecutado por el motor del juego en la fase final.

Es importante indicar que el entorno de diseño de EDoS, tomó como base o pilar 3 modelos existentes: 1. Modelo de objetivos pedagógicos, 2. Modelo de escenarios pedagógicos (IMS-LD-SG) y, 3. El modelo de tareas CTT (*Concur Task Tree*), mismos que se describen brevemente a continuación (Tran et al., 2010):

- 1) **Modelo de objetivos pedagógicos** – Este modelo es considerado un modelo formal específico de dominio de objetivos pedagógicos; mismos vinculados con los objetivos de los juegos serios, es decir, es un modelo vinculado con competencias, conocimientos y comportamientos, entre otros, en el cual se da por validada una competencia cuando se han validado los conocimientos y comportamientos en su totalidad.

El modelo requiere que se haya especificado primeramente los requisitos de los clientes, una vez concluida esta fase, dará inicio el proceso de diseño, cuya primera tarea comienza con la definición de los objetivos pedagógicos del juego.

Para la definición de los objetivos pedagógicos, se utiliza una herramienta que permite a los autores descomponer los objetivos en competencias, mismas que luego se descomponen en conocimientos y comportamientos.

En la [Figura 6.4](#), se puede observar una pestaña para la definición de objetivos pedagógicos en EDoS. Este modelo se considera la base cognitiva del juego serio por desarrollarse.

**Games >> Samoulean >>**

General Information	Design	Production	Tests-Validation	Diffusion		
Objectives	SG Models	Peda - Scenario	Screen Pages	Verification - Validation	Properties	Conditions

General Objectives: Teaching production management methodology Lean

Formation Time: 300h

Target Public: 3rd or 4nd year engineers

**Competences to acquire:**

- C1. Identify and use the global indicators of a company
  - K1 - Diagnose a company (Threshold=100)
  - K2 - Correspondance I.O. / Domain (Threshold=90)
  - K3 - Finding the operation indicators (Threshold=90)
  - K4 - Lean indicators (Threshold=80)
  - B2 - Quality process (2 times)
  - B4 - Collect useful information (2 times)
  - B5 - Use the pertinent tool (1 times)
  - B6 - Respect the confidentiality of information (1 times)
- C2 - Know how to do a general Lean analysis (Threshold 1)
  - K5 - Hierarchically organization of the company (Threshold=72)
  - K6 - BCG matrix (Threshold=100)
  - B1 - Follow the instructions (1 times)
  - B2 - Quality process (1 times)
  - B3 - Organize one's actions (2 times)
  - B4 - Collect useful information (2 times)
  - B5 - Use the pertinent tool (1 times)
  - B6 - Only bother people when useful (1 times)
  - B7 - Take initiative (1 times)
  - B8 - Respect the confidentiality of information (1 times)
- C3 - Know how to do a general Lean analysis (Threshold 2)
- C4 - Manage a deep investigation
- C5 - Establish a sector-based diagnose to determine the operational objectives

**Knowledges:**

- K1 - Diagnose a company
- K2 - Correspondance I.O. / Domain
- K3 - Finding the operation indicators
- K4 - Lean indicators
- K5 - Hierarchically organization of the company
- K6 - BCG matrix

**Behaviours:**

- B1 - Follow the instructions
- B2 - Quality process
- B3 - Organize one's actions
- B4 - Collect useful information
- B5 - Use the pertinent tool
- B6 - Only bother people when useful
- B7 - Take initiative
- B8 - Respect the confidentiality of information

**Figura 6.4** Pestaña para la definición de objetivos pedagógicos en EDoS

Fuente: Figura tomada del texto de (Tran et al., 2010)

Luego de concluir la fase de definición o especificación de los objetivos pedagógicos, da inicio la fase de especificación de escenarios pedagógicos, los cuales deben permitir alcanzar los objetivos definidos, por ello, esta fase se debe elaborar con precisión. A continuación, se describe el modelo de escenario pedagógico (Tran et al., 2010).

- 2) **Modelo escenario pedagógico (IMS-LD-SG)** - Es un modelo basado y derivado de una extensión de la especificación IMS-LD (IMS Learning Design) (IMS, 2003),

la cual ha sido adaptada para construir o diseñar un modelo específico para juegos serios, en este caso, para apoyar en el diseño de escenarios pedagógicos. Según los autores se utilizó IMS-LD como base para el modelo IMS-LD-SG porque permite:

- ✓ IMS-LD es pedagógicamente neutral, compatible con todas las teorías de aprendizaje y no requiere de la definición de un enfoque pedagógico específico.
- ✓ IMS-LD permite diseñar juegos con actividades reutilizables, lo cual permite disponer de una base de datos con componentes de juegos serios reutilizables.
- ✓ IMS-LD permite y garantiza una independencia entre el escenario pedagógico de un juego serio y los recursos informáticos utilizados para su ejecución (memoria, capacidad de almacenamiento, aplicaciones web, componentes propios del juego serio, entre otros). Es decir, se realiza una definición por separado de las actividades pedagógicas y los recursos informáticos. Esta independencia favorece la reutilización y adaptabilidad del escenario pedagógico.
- ✓ IMS-LD además de la organización de actividades, toma en cuenta las actividades de apoyo de los docentes. Además, permite la generación de situaciones educativas con múltiples estudiantes, tal como el trabajo colaborativo en equipos.
- ✓ IMS-LD facilita la personalización y configuración de secuencias e interacciones acordes a los objetivos de aprendizaje del estudiantado. Mismas utilizadas para guías las actividades de aprendizaje y mantener un registro o control de los resultados obtenidos por cada estudiante.
- ✓ IMS-LD permite crear de forma independiente la definición de actividades y su locación (lugar y momento). Las actividades son independientes a los lugares y momentos en los cuales se realizan dentro del escenario. Lo cual permite a los autores cambiar la forma de realización de las actividades

dentro del escenario, sin que se vean afectadas las definiciones de las actividades. Además, permite modificar la navegación de actividades entre dinámicas y estáticas.

El modelo IMS-LD-SG se compone de actividades pedagógicas al igual que IMS-LD, las cuales son el núcleo del modelo y se desarrollan de dos formas: Actividades de aprendizaje y actividades de apoyo o soporte, mismas que pueden ser realizadas desde 2 roles distintos; estudiantes y profesores.

El modelo de escenario pedagógico requiere de la especificación de las relaciones entre los objetivos del escenario y los objetivos pedagógicos (definidos previamente en el modelo de objetivos pedagógicos), igualmente, se debe especificar el tipo de juego serio que se va a diseñar o utilizar. Luego, los objetivos se dividen en actividades que permitan su alcance. Posteriormente, los autores relacionan cada actividad pedagógica con algún recurso computacional (aplicación web o un componente de juego).

Una vez descritos los contenidos pedagógicos del juego, se requiere agregar elementos y características vinculadas con la diversión y jugabilidad (personajes, escenarios divertidos, entre otros).

3) **Descripción de pantallas y game-play** – Se conoce como el modelo de árboles de tareas concurrentes (CTT, *Concur Task Tree*) (Paternó, 1997), el cual es utilizado en cada pantalla del juego serio para formalizar el escenario de la Interacción Persona-Ordenador (IPO) o bien, en inglés HCI (*Human-Computer Interaction*).

A diferencia de los modelos anteriores, los cuales describen y representan aspectos vinculados con pedagogía y el tipo de juego serio por desarrollarse, el modelo CTT se enfoca exclusivamente, a los aspectos del juego serio en sí. Luego de la especificación de las actividades pedagógicas del escenario, se debe realizar una descripción detallada de cada una de las actividades. Para ello, se propone un mecanismo que sirve de ayuda para describir las pantallas con las cuales los usuarios deben interactuar posteriormente.

Las pantallas deben incluir tanto elementos como escenarios divertidos para la Interacción Persona-Ordenador (IPO o HCI), lo cual ayuda a capturar la atención, motivación e inspiración del estudiantado. En otras palabras, EDoS se convierte en un aliado estratégico de los autores para el diseño de pantallas y la especificación de los escenarios IPO de una forma más visual y formal. El uso del modelo CTT permite generar parte del código automáticamente y, luego de completar todos los componentes, se procede a ensamblar los recursos computacionales (aplicación web o un componente de juego).

Finalmente, este modelo permite a los autores insertar aspectos lúdicos en los contenidos pedagógicos del juego serio desarrollado, por tanto, se debe ser cuidadoso en relación con las elecciones que se realicen en este sentido.

**Tabla 6.7 Resumen - Aspectos de análisis de EDoS**

Aspectos generales	País Nivel educativo	Francia Educación superior
Aspectos de diseño	Fundamentación Requerimientos Definición de roles Patrones de diseño Reutilización de software Documentación Diseño de prototipos	DCU No Ingenieros y diseñadores (principalmente). No Sí Sí No
Aspectos metodológico-pedagógicos	Destinatarios Intención pedagógica Objetivos pedagógicos/lúdicos	Docentes y Técnicos (desarrolladores) Formación de habilidades en ingeniería Sí, Descomposición de objetivos en competencias
Aspectos de análisis	Validación de objetivos Evaluación de UX Resultados cuantificables	No No No

#### 4.11.3. Metodología LEGADEE

En **Francia** (Marfisi-Schottman, 2012; Marfisi-Schottman et al., 2010), investigadores de la Universidad de Lyon y Universidad de Grenoble, desarrollaron de forma conjunta una investigación con el objetivo de detallar el proceso de diseño a seguir durante el desarrollo de juegos serios orientados a fines académicos, a la cual le han denominado el nombre de **LEGADEE** (*LEarning GAmes DEsign Environment*). En dicho proceso, realizan una enumeración de los distintos actores que intervendrían durante el diseño y desarrollo del juego serio; gerente de

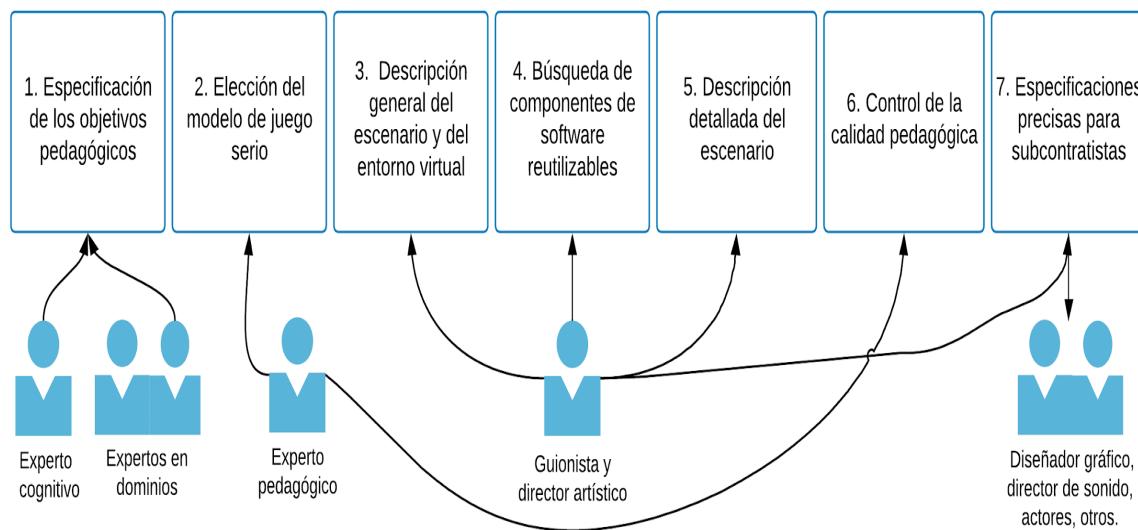
proyecto, entre otros. Luego, los autores proponen una serie de herramientas de especialista cognitivo, expertos en dominios, guionista, director artístico, experto pedagógico, programadores de aprendizaje electrónico (interfaces compuestas por diferentes tipos de *widgets*) adaptadas para el diseño de juegos serios, mismas que permiten especificar las habilidades a potenciar y a construir el escenario pedagógico.

El proceso LEGADEE consta de 7 pasos que sirven de guía para llevar adelante el desarrollo de juegos serios para potenciar la formación de competencias profesionales. La guía se encuentra adaptada a formatos estandarizados y, específicos para el desarrollo de software. Asimismo, en varios de los pasos propuestos, se busca que se reutilice componentes de software y que se garantice la intensión pedagógica del juego. Los pasos se describen a continuación:

- 1) **Especificación de los objetivos pedagógicos** – Debido a que las habilidades por potenciar se pueden organizar en piezas de conocimiento y comportamiento, se debe obtener los conocimientos específicos del dominio que deben asimilar las personas jugadoras. Para ello, se debe consensuar un trabajo en equipo por parte del experto cognitivo y los expertos en dominio con el propósito de identificar los conocimientos y comportamientos importantes para el dominio (se puede utilizar herramientas para la gestión del conocimiento). Además, el experto pedagógico es el responsable de identificar y organizar las habilidades destacadas y definir los objetivos pedagógicos a potenciar a través del juego.
- 2) **Elección del modelo de juego serio** – El experto pedagógico (antes de crear el escenario) realiza la elección de un tipo de modelo predefinido para el diseño del GS (juegos de mesa, investigación, de aventura o rompecabezas). Posteriormente, se debe realizar la adaptación de las herramientas y los distintos módulos de ayuda que se ofrecerán al diseñarse los escenarios divertidos.
- 3) **Descripción general del escenario y del entorno virtual** – Para estructurar el escenario pedagógico y su combinación con un escenario divertido, se debe consensuar un trabajo en equipo por parte del escritor del guion gráfico y del director artístico, debido a que se debe realizar la descripción de los elementos del entorno virtual (personajes, historia, locación de desarrollo de la historia). Se

propone la utilización de herramientas de vista múltiples (interfaces ergonómicas que les permiten interacciones de arrastrar y soltar), es decir, interfaces compuestas por *widgets* (de documentación, recursos, características, lugares, escenarios y, pantallas, entre otras) que se puedan intercambiar, mover, adaptar y, utilizar en las diferentes dimensiones del juego serio.

- 4) **Búsqueda de componentes de software reutilizable** – Debido a que es más eficiente reutilizar componentes, a tener que diseñar nuevamente a todos los elementos, se recomienda a los diseñadores indagar previamente en la base de datos, si existe algún elemento (previamente desarrollado para otro proyecto) que pueda satisfacer las necesidades requeridas en el diseño del juego serio.
- 5) **Descripción detallada del escenario** – Posteriormente a que se han integrado al escenario los componentes de software reutilizables, se debe realizar por parte del escritor del guion gráfico y del director artístico la descripción de las partes faltantes. Además, detallar la ilustración de las escenas tomando en cuenta los detalles e interacciones para que sean integradas al juego por los programadores.
- 6) **Control de la calidad pedagógica** – Se realiza una fase de evaluación previa antes de la producción del juego. Con el propósito de asegurarse de que no haya caminos sin salida y garantizar el alcance de los objetivos pedagógicos, es recomendable realizar un conjunto de pruebas iniciales en el gráfico de escenarios del juego serio. Luego, para pruebas más exhaustivas, se podrían utilizar jugadores virtuales, con el objetivo de evaluar estadísticamente si el juego serio cumple con su intención pedagógica.
- 7) **Especificaciones precisas para subcontratistas** – Este punto hace referencia a que, antes de pasar a la fase de producción, se debe completar las especificaciones de cada subcontratista (diseñador gráfico, gerente de sonido, actores, entre otros), responsabilidad que recae sobre el director artístico. La [Figura 6.5](#), resume los pasos propuestos y descritos anteriormente.

**Figura 6.5** Proceso de 7 pasos para el diseño de SG, LEGADEE

**Fuente:** Elaboración propia a partir del texto de (Marfisi-Schottman, 2012; Marfisi-Schottman et al., 2010)

En suma, en la [Figura 6.5](#), se aprecia que el primer paso del proceso es definir los objetivos pedagógicos (Evans et al., 2016, p. 4). Luego, se proponen evaluaciones a nivel pedagógico, sin embargo, existe una ausencia de evaluaciones a nivel del juego y la jugabilidad de este.

**Tabla 6.8** Resumen - Aspectos de análisis de LEGADEE

Aspectos generales	País Nivel educativo	Francia Educación superior
Aspectos de diseño	Fundamentación Requerimientos  Definición de roles  Patrones de diseño Reutilización de software Documentación Diseño de prototipos	No indica No Gerente de proyecto, especialista cognitivo, expertos en dominios, guionista, director artístico, experto pedagógico, programadores.  No Sí Múltiples herramientas No
Aspectos metodológico-pedagógicos	Destinatarios  Intención pedagógica  Objetivos pedagógicos/lúdicos	Docentes y diseñadores Formación de competencias profesionales Sí
Aspectos de análisis	Validación de objetivos Evaluación de UX Resultados cuantificables	Sí No No

En suma, como resultado de esta investigación se ofrece una serie de recomendaciones de como guiar un proceso de diseño de juegos serios.

#### 4.11.4. Metodología SAVIE

En **Canadá** (Sauvé, 2009), el centro para la experiencia y la investigación en el aprendizaje permanente (**SAVIE**), desarrollaron un modelo de diseño pedagógico interactivo para el desarrollo de juegos serios que permitan potenciar el aprendizaje en línea, específicamente, fue pensado para ser utilizado por el profesorado en los diferentes niveles educativos (primaria, secundaria, educación técnica y superior).

Este modelo de diseño pedagógico interactivo está compuesto por 6 esquemas genéricos (*Shells*) para la construcción de juegos serios educativos en línea, el diseño tiene como intención pedagógica el aprendizaje, especialmente, la educación a distancia.

El modelo permite al profesorado crear distintos juegos serios partiendo de una misma base o estructura, los cuales pueden ser accedidos por el estudiantado a través de internet. Por ejemplo, se utilizó para modificar la estructura original del juego “Parcheesi”, es decir, para modificar algunas de las reglas del juego con el propósito de hacerlo más fácil de utilizar y agregar más actividades de aprendizaje. Además, se utilizó Parcheesi para desarrollar otro juego en línea que permite al jugador adquirir conocimientos vinculados con la prevención del asma.

Las herramientas ofrecidas al personal docente pueden ser utilizadas para (Sauvé, 2009):

- ✓ Fijar los valores de los parámetros del juego.
- ✓ Generar las reglas para definir los movimientos de los jugadores.
- ✓ Crear el material educativo/pedagógico.
- ✓ Definir los criterios para fijar el fin del juego y determinar ganadores.
- ✓ Elaborar las herramientas requeridas para revisar y evaluar el juego.

Según la autora (Sauvé, 2009), estas herramientas permiten que el juego se actualice de forma continua y, así, garantizar su impacto de aprendizaje durante el proceso formativo. Igualmente agrega que, un marco de juegos (*framework*) es un juego ya existente, al cual se le borran los contenidos para obtener únicamente su estructura básica, e indica que todo juego se puede dividir en 2 componentes principales; la estructura de juego y los contenidos.

- ✓ **La estructura del juego** - Se determina la forma en que se juega, la cual incluye la definición de reglas, los pasos a seguir durante el juego, movimientos de jugadores, retos que deben enfrentar los jugadores y estrategias que se deben utilizar para ganar. Cuando la estructura se ha definido y analizado claramente, se convierte en un Marco o Shell de juego genérico, listo para ser programado y colocarse en línea.
- ✓ **Los contenidos** - Se define la información a transmitir durante la ejecución del juego, esta información está vinculada con los objetivos pedagógicos que se desean alcanzar y las competencias a potenciar mediante su interacción.

Para la creación o diseño de la consola de juegos educativos genéricos en línea, (Sauvé, 2009) ha desarrollado el modelo de diseño pedagógico interactivo, el cual está conformado por 5 fases:

- 1) **Análisis** - Se realiza el análisis de la población estudiantil al que está dirigido el juego, igualmente, se analiza el contexto en el que se desarrollará acción formativa; especificar los requisitos pedagógicos y tecnológicos del Shell, revisar los marcos de juegos existentes y seleccionar la estructura del juego a adaptar, se identifican los elementos de composición del Shell, en función de la estructura y los contenidos del juego existente.
- 2) **Diseño** - Se identifica la estructura y los elementos de contenido del juego que deben ser modificados para crear el Shell, luego, se deben describir los elementos del Shell y sus funciones, además, se realiza el diseño de un modelo visual usando imágenes de pantalla del Shell, sus variantes y escenarios.

- 3) **Desarrollo técnico** - Se realiza la redacción de las especificaciones técnicas, mismas que deben contener los principios para la construcción en línea; desarrollo gráfico y multimedia del Shell, además, se realiza la programación de diferentes elementos y sus funciones en el Shell, finalmente, se realizan las pruebas de integración funcional del Shell con los contenidos de un juego educativo.
- 4) **Evaluación formativa del juego genérico** - Se realiza la especificación de los criterios y procesos de evaluación formativa; se desarrollan los instrumentos de evaluación para los usuarios del juego y, se ejecutan las revisiones que sean necesarias.
- 5) **Evaluación resumida de los juegos creados con el Shell genérico del juego** - Se desarrolla un juego educativo a través del Shell, se realizan las especificaciones del marco experimental, se desarrollan los instrumentos de medición para ser utilizados por expertos y la población usuaria, se valida el juego por expertos, se lleva a cabo las pruebas del juego por parte de los usuarios, finalmente y, de ser necesario, se vuelve a efectuar una revisión del juego y el Shell.

En suma, la estructura y el contenido del juego deben mantener un equilibrio, para poder reconocer e intercambiar material educativo. Como resultado se propone una plantilla de juego y un método que permiten realizar la adaptación de diferentes contenidos educativos a través de la plantilla, a modo de ejemplo se indica que la plantilla puede ser utilizada por el profesorado para cargar contenido relacionado con el área de la salud.

**Tabla 6.9 Resumen - Aspectos de análisis de SAVIE**

Aspectos generales	País Nivel educativo	Canadá Primaria, secundaria, educación terciaria y superior
Aspectos de diseño	Fundamentación Requerimientos Definición de roles Patrones de diseño Reutilización de software Documentación Diseño de prototipos	DCU Sí No No No Sí Sí
Aspectos metodológico-pedagógicos	Destinatarios Intención pedagógica Objetivos pedagógicos/lúdicos	Docentes y estudiantes Aprendizaje; educación a distancia Sí

Aspectos de análisis	Validación de objetivos Evaluación de UX Resultados cuantificables	Sí Sí No
----------------------	--	----------------

#### 4.11.5. Modelo DODDEL

En **Nueva Zelanda** (McMahon, 2009), investigador de la Universidad de Auckland, propone el Modelo Diseño Orientado a Documentación y Desarrollo de Aprendizaje Experimental (**DODDEL**, *Document-Oriented Design and Development of Experiential Learning*), el cual es una metodología orientada para ser utilizada por diseñadores y desarrolladores, con la cual pueden producir documentación y establecer guías para ayudar al diseño y desarrollo de juegos serios. El modelo consta de 4 fases de desarrollo y un nivel de evaluación con el propósito de equilibrar los diseños de los juegos.

El modelo DODDEL, es una metodología centrada en documentación la cual guía o conduce el proceso completo del ciclo de vida de desarrollo de videojuegos. DODDEL al igual que la metodología EMERGO de (Nadolski et al., 2008), toma como base de referencia las etapas tradicionales de diseño y desarrollo ADDIE (Análisis, Diseño, Desarrollo, Implementación y Evaluación).

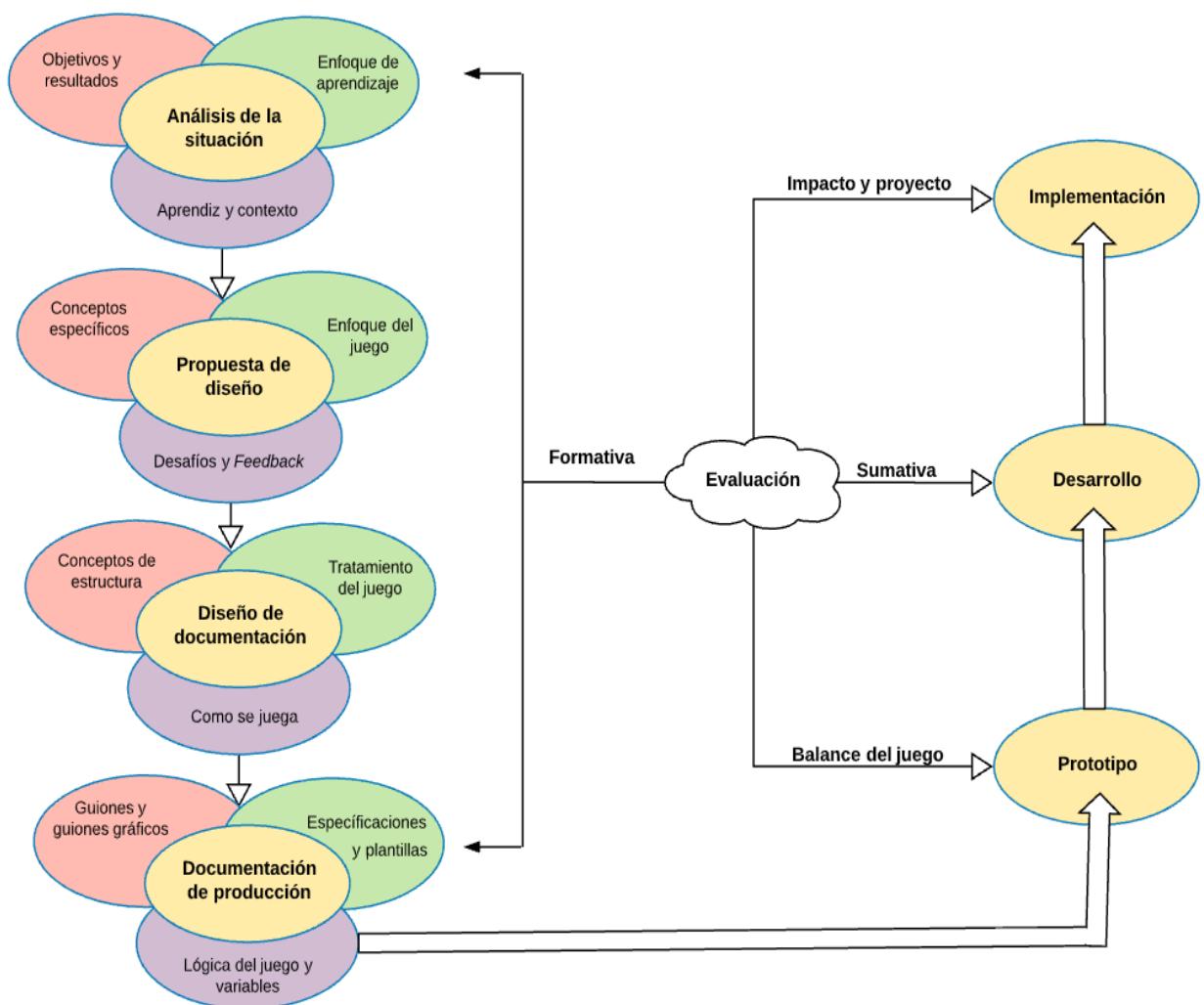
Se indica que DODDEL posee características clave que lo diferencian de otros modelos tradicionales (McMahon, 2009):

- ✓ Una articulación desde un amplio enfoque de diseño hasta especificaciones detalladas.
- ✓ Múltiples iteraciones dentro de cada etapa que incorporan elementos dependientes específicos de los resultados de aprendizaje, las necesidades y características del usuario, y la estrategia de aprendizaje desarrollada.
- ✓ Heurística para guiar el contenido dentro de cada elemento.
- ✓ Documentos específicos de resultados en cada etapa del proceso.

El modelo DODDEL se estructura en 4 fases, mismas que se citan y describen brevemente a continuación (McMahon, 2009):

- 1) **Análisis de la situación** - En esta fase se requiere que los diseñadores articulen una filosofía de aprendizaje que sea congruente con el producto apropiado. Se sugiere considerar 3 componentes: Objetivos y resultados, aprendiz y contexto y, el enfoque de aprendizaje.
- 2) **Propuesta de diseño** - En esta fase se sugiere que los diseñadores consideren 3 componentes; desarrollen los conceptos específicos y objetivos, los desafíos y el *Feedback* (la estrategia de aprendizaje) y, la propuesta o enfoque del juego, entre otros.
- 3) **Diseño de documentación** - En esta fase se sugiere considerar 3 componentes; Conceptos de estructura, como se juega y, el tratamiento del juego.
- 4) **Documentación de producción** - En esta fase se requiere que se forme la base de un acuerdo contractual entre clientes, desarrolladores, expertos en la materia, entre otros. Se sugiere considerar 3 componentes: Guiones y guiones gráficos (historia), variables y la lógica del juego, las plantillas generales y las especificaciones.

El modelo DODDLE afronta todo el ciclo de vida del desarrollo de sistemas e igualmente, propone una evaluación central (acorde a los modelos de diseño instruccional), sin embargo, se incorpora una nueva etapa de creación de prototipos con el propósito de equilibrar/balancear el diseño del juego; este nivel de equilibrio se agrega como otro nivel de evaluación, que va más allá de la evaluación tradicional formativa, sumativa y de impacto (McMahon, 2009). En la [Figura 6.6](#), se puede apreciar el modelo DODDLE con sus 4 fases de desarrollo y, el nivel de evaluación propuesto.



**Figura 6.6** *Modelo DODDEL*

Fuente: Elaboración propia a partir del texto de (McMahon, 2009)

**Tabla 6.10** Resumen - Aspectos de análisis de DODDEL

Aspectos generales	País Nivel educativo	Nueva Zelanda Educación terciaria y superior
Aspectos de diseño	Fundamentación Requerimientos Definición de roles Patrones de diseño Reutilización de software Documentación Diseño de prototipos	ADDIE No Diseñadores y desarrolladores No No En cada etapa del proceso Sí, para equilibrar el diseño del juego
Aspectos metodológico-pedagógicos	Destinatarios Intención pedagógica Objetivos pedagógicos/lúdicos	Diseñadores y desarrolladores Guia el proceso de diseño de juegos serios Sí
Aspectos de análisis	Validación de objetivos Evaluación de UX	No Sí, A través de los prototipos

#### 4.11.6. Metodología VGSL

En **España** (Padilla-Zea, 2011), investigadora de la Universidad de Granada propone un proceso para el desarrollo de videojuegos educativos, en el cual se considera un equilibrio entre tres factores; el aprendizaje, la diversión y actividades de aprendizaje colaborativas, al cual le asignó el nombre de Aprendizaje Colaborativo Soportado por Videojuegos (**VGSL**, *Video Game - Supported Collaborative Learning*). El modelo fue pensado para que el profesorado pueda crear juegos serios que permitan potenciar competencias para el aprendizaje colaborativo. Luego, la autora señala que los componentes principales de un sistema VGSL corresponden a: los objetivos educativos y lúdicos, tareas educativas y lúdicas e interacción. Para organizar la información necesaria en el sistema VGSL, se han presentado cuatro conjuntos de modelos, que permiten caracterizar el proceso de aprendizaje, el proceso de juego, la relación entre dichos procesos y al usuario, además, permitir un proceso de adaptación a distintos niveles por medio de la configuración de sus parámetros. La metodología VGSL, responde a dos objetivos (Padilla-Zea, 2011, p. 157):

- ✓ Proporcionar a los profesores una herramienta con la que los estudiantes puedan alcanzar los objetivos educativos de forma atractiva.
- ✓ Proveer un sistema de monitorización del proceso de aprendizaje que están desarrollando los alumnos con objeto de permitir adaptaciones en el proceso de juego que repercutan en una mejora de los resultados de aprendizaje.

Para alcanzar el cumplimiento de los objetivos, la autora (Padilla-Zea, 2011) recomienda modelar al estudiantado desde 3 perspectivas: aprendizaje, juego e interacción con otros compañeros.

El modelo VGSL fue diseñado pensando en una aproximación basada en modelos, la cual permite ejecutar abstracciones explícitas del sistema independientes de la implementación posterior, lo cual contribuye a una mejor

flexibilidad, que luego facilitará el mantenimiento y reutilización del sistema, así como de la información almacenada en estos modelos. Entre las ventajas de la utilización de la aproximación basada en modelos están (Padilla-Zea, 2011):

- ✓ Alcanzar un alto nivel de representación que permitirá identificar las fuentes de información y estructurar dicha información.
- ✓ Adquirir conocimientos de una forma guiada.
- ✓ Diseñar de forma genérica independientemente del dominio.
- ✓ Utilizar distintos entornos de desarrollo para la implementar el sistema diseñado.

Para alcanzar el objetivo propuesto vinculado con las adaptaciones durante el proceso formativo que permitan una mejora en los resultados obtenidos por parte del estudiantado, se propone dividir en 4 grupos el modelo propuesto:

- ✓ Modelos para definir y monitorizar el proceso educativo.
- ✓ Modelos para especificar y monitorizar el contenido lúdico.
- ✓ Modelos para relacionar el contenido educativo y el lúdico, lo cual va a permitir definir cómo los contenidos lúdicos satisfacen los requisitos educativos.
- ✓ Modelos de usuarios para monitorizar la evolución educativa del estudiantado mientras juegan.

La metodología VGSC se utilizó para desarrollar el juego serio “*Nutri-Galaxy*” el cual permite a estudiantes de secundaria (de 11 a 12 años) potenciar el aprendizaje colaborativo (juego, educación y colaboración) y la adquisición de competencias a nivel general (Padilla-Zea et al., 2015).

**Tabla 6.11 Resumen - Aspectos de análisis de VGSC**

Aspectos generales	País Nivel educativo	España Inicial, primaria y secundaria
--------------------	-------------------------	--

Aspectos de diseño	Fundamentación Requerimientos Definición de roles Patrones de diseño Reutilización de software Documentación  Diseño de prototipos	DCU No Docentes y estudiantes Sí, Patrones de aprendizaje y tutoría Sí No Sí, para la definición de Objetivos Educativos; valida la utilidad del proceso de diseño.
Aspectos metodológico-pedagógicos	Destinatarios Intención pedagógica Objetivos pedagógicos/lúdicos	Docentes y diseñadores Aprendizaje colaborativo (juego, educación y colaboración) Sí
Aspectos de análisis	Validación de objetivos Evaluación de UX Resultados cuantificables	Sí, Sistema de monitoreo del proceso de aprendizaje. Sí, Mediante entrevistas y cuestionarios Sí, Mediante entrevistas y cuestionarios

#### 4.11.7. Metodología MECONESIS

En **Colombia** (Cano, 2016), investigadora de la Universidad del Cauca, propone una metodología para guiar el diseño de juegos serios, la cual fue denominada **MECONESIS** (**M**Etodología para la **C**ONcepción de ju**E**gos **S**erios para nIñoS con discapacidad auditiva). Toma como referencia el Proceso Unificado de Desarrollo de Software (UP) (Kroll & Kruchten, 2003). Para el modelado de las interacciones utiliza las notaciones de CTT (*Concurrent Task Trees*), asimismo, utiliza UML para modelar los diagramas de clases, además, para la descripción de los escenarios hace uso de meta-datos como IMS-LD y BMPN para la descripción de los procesos.

La metodología se encuentra estructurada en **4 fases** (Análisis, Pre-producción, Producción y Post-producción) y **7 modelos** (modelo de análisis, modelo de usuario, modelo de adaptación, modelo de objetivos pedagógicos, modelo de tareas, modelo de escenarios y modelo de validación). A continuación se describen brevemente los 7 modelos (Cano, 2016; Cano et al., 2016; Garita et al., 2019):

- 1) **Modelo de análisis** - Este modelo se enfoca en la evaluación de la experiencia del usuario, permite la identificación de problemas y aspectos de diseño, usuario

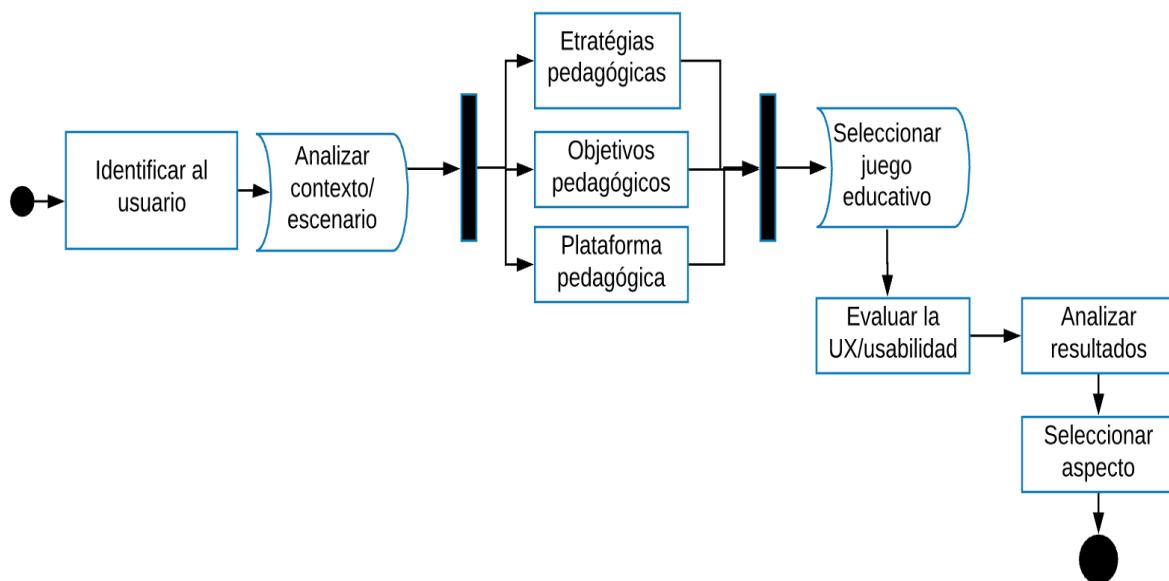
y del juego como tal (desarrollo). Entre los factores que comprende este modelo están: estudios observacionales, estrategias pedagógicas/lúdicas, escenarios pedagógicos/lúdicos, validación y comunicación. El análisis de la experiencia del usuario permitirá tomar en cuenta ciertos aspectos relevantes en el modelo de usuario.

- 2) **Modelo de usuario** - En este modelo se analizan los aspectos identificados a partir de la experiencia del usuario, mismos que servirán de apoyo para ajustar el juego a los requerimientos identificados. Es decir, se utiliza para determinar los aspectos del usuario, para luego adaptar los aspectos del juego, tomando en cuenta sus necesidades, intereses y comportamientos).
- 3) **Modelo de adaptación** - En este modelo se valoran y toman en cuenta las características del usuario, mismas que son requeridas para realizar las adaptaciones de los elementos del juego acorde a los procesos cognitivos. Por ello, se le relaciona con el modelo de objetivos pedagógicos.
- 4) **Modelo de objetivos pedagógicos/lúdicos** - En este modelo basado en competencias donde se incluye la participación del profesorado, el cual determina las competencias que se desean alcanzar con el usuario final. El modelo se debe incluir en la fase inicial y en la fase final de la metodología, lo cual permitirá posteriormente, validar los alcances esperados.
- 5) **Modelo de tareas** - En este modelo se definen las tareas por realizar en el juego, tales como tareas vinculadas con el sistema, interacción y usuarios. Para alcanzar este propósito, el modelado se puede realizar a través del modelo CTT.
- 6) **Modelo de escenarios** - En este modelo se deben considerar los escenarios pedagógicos y lúdicos.
- 7) **Modelo de validación** - En este modelo se realiza la evaluación de los diferentes aspectos de producción del juego, además, permite contrastar el cumplimiento de los objetivos pedagógicos establecidos en la fase inicial.

A continuación se describen las 4 fases (Cano et al., 2016; Garita et al., 2019):

- 1) **Fase de análisis** - Esta fase corresponde a la primera fase de análisis de requerimientos en la ingeniería de software. En esta etapa la autora

recomienda que se debe incluir desde el inicio la participación del usuario final y, fundamentada la fase en la metodología de Diseño Centrada en el Usuario para Sistemas Interactivos ISO 13407 (ISO/IEC, 1999), con el propósito de definir los requerimientos en el escenario pedagógico y lúdico. La fase incluye varias etapas encargadas de la identificación de los distintos actores que forman parte del equipo de producción del juego serio; se incluye al usuario, producto, actividad y contexto de uso. Lo anterior, para explorar e identificar factores humanos de los usuarios (no todos los usuarios tienen el mismo nivel de experiencia del juego ni aprenden al mismo ritmo. Se consideran 3 modelos: **modelo de análisis, modelo de objetivos pedagógicos/lúdicos y el modelo de usuario**. La Figura 6.7, permite observar gráficamente el proceso de análisis para la evaluación de las experiencias del usuario.



**Figura 6.7 Proceso de análisis para evaluar la experiencia del usuario - MECONESIS**

Fuente: Elaboración propia a partir del texto de (Cano, 2016, p. 209)

2) **Fase de pre-producción** – Esta fase se vincula con el diseño de la interfaz del juego, donde se incluyen patrones de diseño para las interfaces del juego y para la implementación y las guías necesarias para usuarios. Los factores

involucrados en la identificación de elementos están: estudio de observación, las estrategias pedagógicas/lúdicas, escenarios, implementación, validación y comunicación (lenguaje, dibujos, feedback, ayuda, mensajes de motivación, entre otros). Además, intervienen los modelos de tareas y escenarios, mismos vinculados al contenido del juego. En suma, esta fase permite identificar aspectos que pueden ser tomados en cuenta para el diseño.

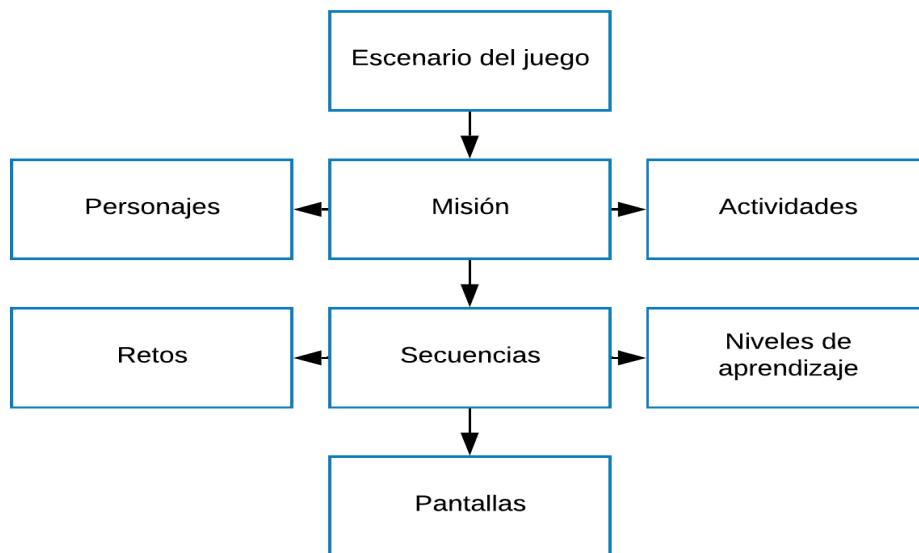
3) **Fase de producción** - Esta fase se vincula con la implementación del juego serio. Está más orientada para el uso del desarrollador, debido a que integra componentes necesarios para su funcionamiento, el cual será el encargado de desarrollar el juego según las vistas y modelos de los escenarios propuestos por el diseñador (por ello, debe existir una clara comunicación entre ambos). A través del modelo de tareas debe obtener los requerimientos funcionales y no funcionales. Al momento de llegar a la creación de los modelos UML de la aplicación, se requiere tener claridad en relación con el tipo de desarrollo por implementar (ampliación móvil, web, consola o PC), además, de los elementos y componentes de hardware que se van a requerir para la interacción con el juego por parte del usuario final. Además, la autora (Cano, 2016) recomienda utilizar en ésta fase un modelo de escenarios que podrían contener entre otros, los siguientes elementos: **actividades, historia (personajes, misiones), tipo de juego e interacciones**. El modelo de escenarios podría llegar a incluir diferentes escenarios, dependiendo de la actividad y personajes de la historia.

Según (Cano, 2016, p. 216) cada escenario podría tener la siguiente información:

- ✓ Descripción visual y textual de los elementos de la interfaz y precisar los personajes, los lugares y los objetos presentes en la pantalla.
- ✓ Las indicaciones asociadas a la misión destinada al jugador.
- ✓ Diálogos entre los personajes de la escena.

- ✓ Las acciones que realizará el jugador.

La [Figura 6.8](#), permite observar una propuesta de un modelo de escenario.



**Figura 6.8** *Modelo de escenario de MECONESIS*

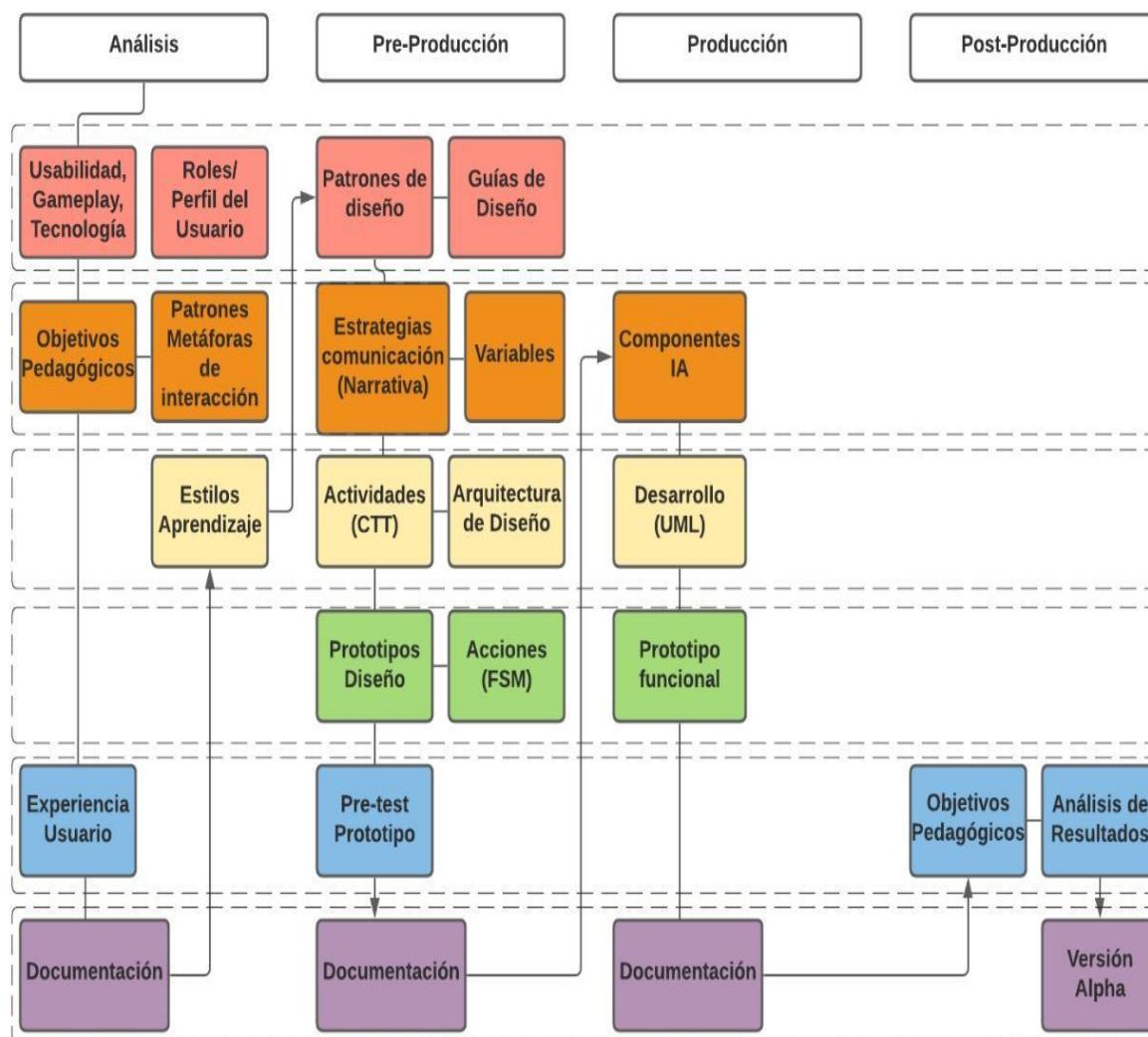
Fuente: Elaboración propia a partir del texto de (Cano, 2016)

- 4) **Fase de post-producción** - Esta fase se encuentra relacionada con la evaluación del juego serio, para ello, se propone un modelo de evaluación del mismo, acá se toma el rol del usuario final y el experto para realizar la evaluación. Esta fase de evaluación permite consolidar varios aspectos del juego y realizar la validación de los alcances de los objetivos propuestos en la fase de análisis. Finalmente, la evaluación propuesta consta de 4 facetas: Objetivos pedagógicos, interacciones, problemas y progresión, finalmente, condiciones de utilización.

Tal como se puede apreciar en la [Figura 6.9](#), la metodología permite la adaptación de los modelos y las herramientas para cada fase, con el propósito de establecer una misma comunicación entre los diferentes actores, los cuales a su vez poseen distintas competencias en el dominio (Cano, 2016).

Finalmente, la metodología MECONESIS se utilizó para desarrollar el juego serio "**Lectoescritura con Fitzgerald**", el cual fue desarrollado para dispositivos

móviles Android para ser utilizado por niños de secundaria (de 12 a 15 años) con discapacidad auditiva.



**Figura 6.9 Metodología MECONESIS**  
**Fuente:** Adaptación propia a partir del texto de (Cano, 2016, p. 209)

**Tabla 6.12 Resumen - Aspectos de análisis de MECONESIS**

Aspectos generales	País Nivel educativo	Colombia Educación especial
Aspectos de diseño	Fundamentación Requerimientos  Definición de roles  Patrones de diseño Reutilización de software Documentación	UP, DCU con CTT, UML, IMS-LD y BMPN En el escenario pedagógico y lúdico. Psicólogo, docentes, investigadores en ciencias computacionales, diseñador, desarrollador y niño con problemas auditivos (12 a 15 años). Patrones de diseño para interfaz gráfica. Sí Dentro de la etapa comunicación

	Diseño de prototipos	Sí, prototipo funcional
Aspectos metodológico-pedagógicos	Destinatarios Intención pedagógica Objetivos pedagógicos/lúdicos	Niños con discapacidad auditiva y docentes. Aprendizaje (lectoescritura) Basado en competencias
Aspectos de análisis	Validación de objetivos Evaluación de UX Resultados cuantificables	Sí, Modelo de validación Sí, Modelo de análisis: Observación, validación y comunicación, estrategias y escenarios pedagógico/lúdicos. Sí, Etapa de validación; cualitativa y cuantitativa.

#### 4.11.8. Metodología MPi+u+a

En **España** (Granollers, 2004), el investigador de la Universidad de Lleida, propone una metodología a la cual denomina Modelo de Proceso de la Ingeniería de la Usabilidad y de la Accesibilidad (**MPi+u+a**), esta metodología está orientada hacia el diseño de sistemas interactivos centrados en el usuario. El modelo propuesto se estructura en 6 fases; análisis de requisitos, diseño, implementación, lanzamiento, prototipado y evaluación.

En la metodología MPi+u+a, se resalta que la claridad de la información y la consistencia de esta, son características de suma importancia para la usabilidad y la accesibilidad del sistema. Asimismo, se indica que, todo método debe disponer de un esquema mediante el cual el o los usuarios del modelo (parte del equipo de desarrollo) identifiquen en el momento de ser necesario la fase del desarrollo en la cual se encuentra y las opciones con las que cuenta a partir de ella para continuar con la fase de desarrollo (Granollers, 2004). Es decir, se propone una integración de la ingeniería del software con los principios de ingeniería de usabilidad y accesibilidad, según el autor, esta integración permitirá generar una metodología idónea para guiar a los desarrolladores a través del proceso de implementación de un sistema interactivo.

Un aspecto que se destaca como importante en la metodología MPi+u+a, es brindar la máxima importancia al usuario final, por ello, esta metodología se fundamenta en el diseño centrado en el usuario. A continuación, se describe brevemente las 6 fases de desarrollo propuestas (Cano, 2016; Granollers, 2004):

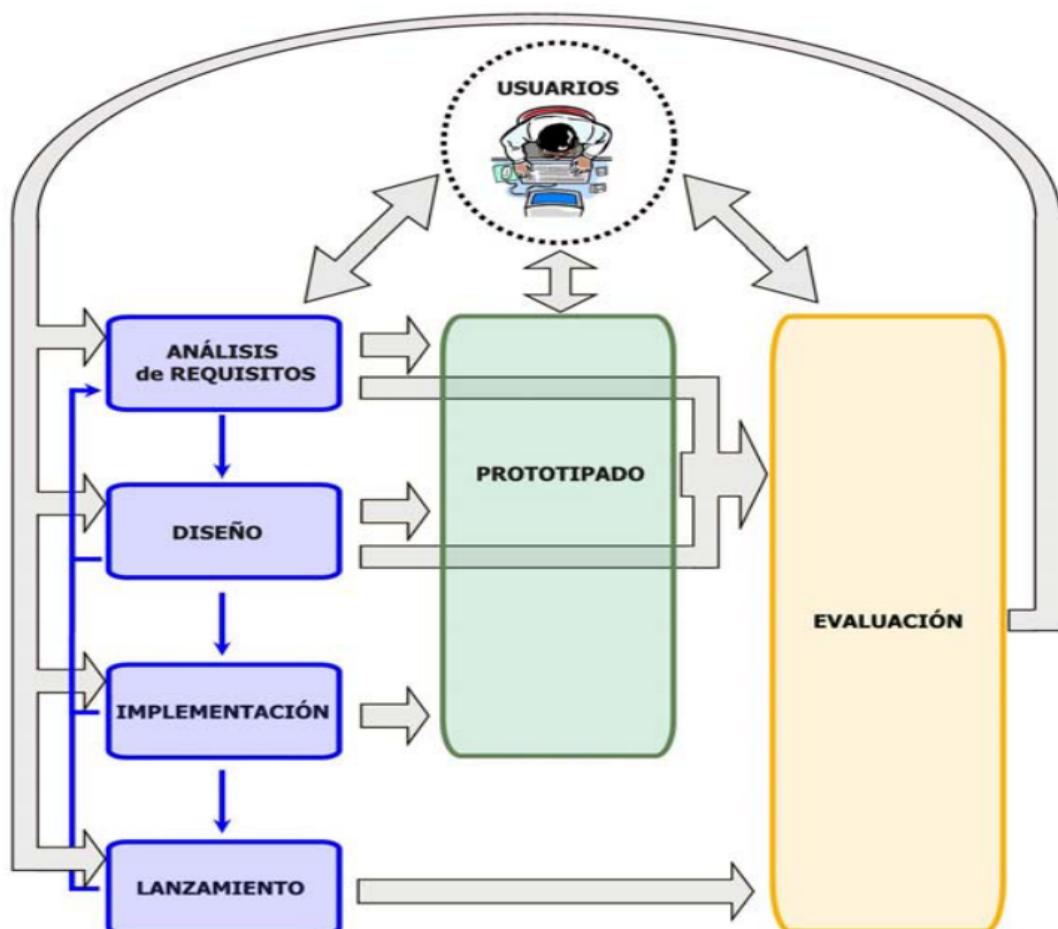
- ✓ **Análisis de requisitos** - En esta fase se considera la comunicación con los usuarios como un aspecto prioritario al desarrollarse un software, máxime que

las necesidades y las experiencias de los usuarios pueden variar. Además, se recomienda tomar en cuenta en esta fase los siguientes factores: análisis etnográfico, implicados (*Stakeholders*), clasificación de los usuarios, objetivos, plataforma y reflexiones vinculadas con la información recolectada.

- ✓ **Diseño** - En esta fase se abordan dos funcionalidades diferentes; el diseño de la actividad (especificación funcional, la tecnología y posibilidades que el sistema ofrece para que las personas sean capaces de utilizar sistemas interactivos para el logro de sus actividades) y diseño de la información (brindar soporte a la percepción, la interpretación y la comprensión de la información). Actividades identificadas como principales para conformar el diseño global de la interacción.
- ✓ **Implementación** - Esta fase es conocida también como la fase de codificación, ya que es la fase donde se debe escribir el código para la implementación del sistema, el cual debe cumplir con todas las especificaciones establecidas en la fase de análisis de requisitos, responda al diseño del sistema y a los objetivos planteados.
- ✓ **Lanzamiento** - En esta fase se recomienda la comprobación de la aceptabilidad del sistema (mediante la combinación de aceptabilidad social y práctica). Igualmente, se recomienda obtener un feedback por parte del usuario final a través de pruebas.
- ✓ **Prototipado (*prototyping*)** – En esta fase el prototipo actúa como un mecanismo para realizar comprobaciones durante todo el desarrollo, el cual se incluye desde la fase inicial de la metodología, donde el usuario final se hace participar durante el desarrollo para lograr evaluar el producto desde la fase inicial (verificar funcionalidad, aspecto de la interfaz, entre otros) y, con ello, recoger las impresiones del usuario para incidir en el diseño de la interfaz final.
- ✓ **Evaluación** - En esta fase se recomienda que la evaluación se realice durante todo el ciclo de vida del proceso de desarrollo, donde los resultados deben

aportar mejoras vinculadas a las soluciones evaluadas y se deben corregir los posibles errores que se reporten. La evaluación se convierte en un aspecto clave para obtención de sistemas interactivos usables y accesibles. Por ende, se recomienda realizar una elección de las técnicas que se crean convenientes para la obtención del feedback por parte del usuario, considerando las métricas de usabilidad y métodos de evaluación.

La Figura 6.10, presenta la metodología MPi+u, acorde con (Granollers, 2004), la metodología fue confeccionada tomando como referencia la codificación de colores orientados a la ingeniería de software (color azul), luego, el prototipado el cual abarca las técnicas que permitirán la posterior fase de evaluación (color verde). Por último, la evaluación, la cual circscribe e incluye métodos para la evaluación (color amarillo).



**Figura 6.10 Metodología MPi+u: integración de la Usabilidad y Accesibilidad**  
**Fuente:** Figura tomada del texto de (Granollers, 2004, p. 159)

**Tabla 6.13 Resumen - Aspectos de análisis de MPJu+a**

Aspectos generales	País Nivel educativo	España Secundaria
Aspectos de diseño	Fundamentación Requerimientos Definición de roles Patrones de diseño Reutilización de software Documentación  Diseño de prototipos	DCU Ingeniería de requisitos - ISO/IEC 9126-1 Usuarios finales No No Fundamentada en ISO 13407 Sí, para verificar funcionalidad, aspecto de la interfaz, recoger las impresiones del usuario.
Aspectos metodológico-pedagógicos	Destinatarios  Intención pedagógica  Objetivos pedagógicos/lúdicos	Desarrolladores Diseño de sistemas interactivos Diseño de sistemas interactivos centrados en el usuario.  Sí
Aspectos de análisis	Validación de objetivos  Evaluación de UX  Resultados cuantificables	Sí, Durante el ciclo de vida del desarrollo. Sí, mediante feedback y a través de pruebas. No

#### 4.11.9. Metodología MPDSG

En **Argentina** (Evans et al., 2016), investigadores de la Universidad Nacional de Mar del Plata, han diseñado el **Modelo de Proceso de Desarrollo para Serious Games (MPDSG)**. Modelo que ha sido utilizado en diferentes investigaciones (Evans et al., 2016; Evans & Massa, 2017; Massa et al., 2017; Spinelli et al., 2017). Esta metodología propone un modelo de procesos que garantiza la jugabilidad e inmersión del jugador.

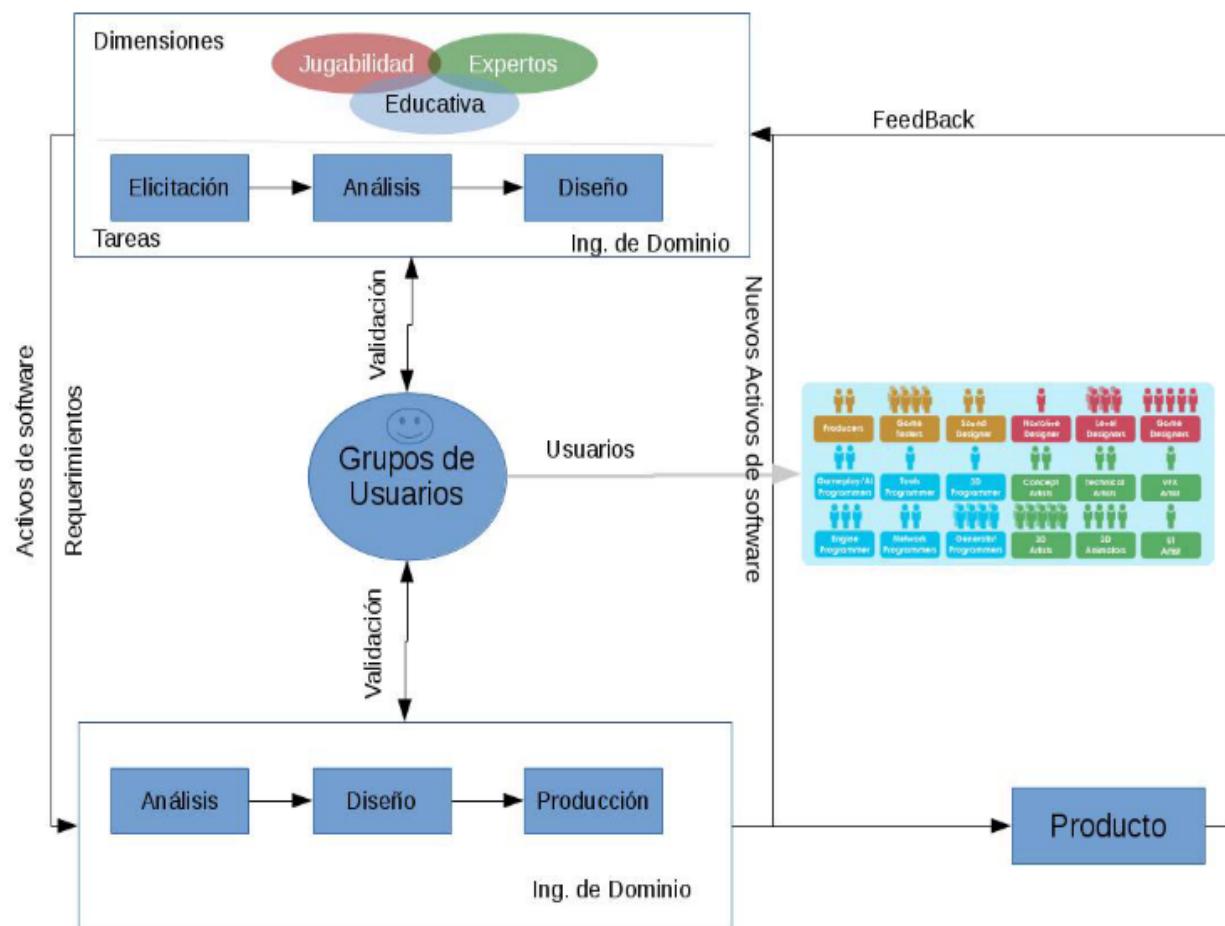
La metodología fue pensada para el desarrollo de juegos serios que permiten potenciar la adquisición de competencias del siglo XXI, por ello, aplicaron la metodología propuesta para crear el juego serio llamado *Power Down the Zombies*, el cual permite a estudiantes de cuarto año de secundaria evaluar el impacto medioambiental y social del uso tecnológico de la energía y, a su vez, permite reflexionar críticamente acerca del uso que se les está dando a los recursos naturales.

La metodología MPDSG toma como referencia las Líneas de Producción de Software (LPS) (Clements & Northrop, 2001), el proceso de Diseño Centrado en el Usuario (DCU) (Hassan-Montero & Ortega-Santamaría, 2009) y el Modelo de Proceso para el Desarrollo de Objetos de Aprendizaje (MPOBA) (Massa, 2013), con

el propósito de satisfacer los objetivos de jugabilidad e inmersión. Los puntos más importantes que han sido considerados de cada proceso son (Evans et al., 2016):

- ✓ Con respecto a la metodología de **LPS**, se aplica como marco general para la construcción del juego, hace un fuerte énfasis en la reusabilidad (*core assets*).
- ✓ Con respecto al enfoque **DCU**, se validan los atributos de jugabilidad e inmersión.
- ✓ Con respecto al modelo de proceso **MPOBA**, se utiliza para organizar y validar el correcto diseño de los elementos de aprendizaje, considerando los *assets* y los juegos serios como un Objeto de Aprendizaje (OA).

En el Modelo MPOBA se destaca que los expertos que deben estar involucrados en el desarrollo de un juego serio son: los expertos en jugabilidad, los expertos en pedagogía (docentes) y los expertos del contenido o dominio que atraviesa el videojuego. Lo anterior, con el propósito de que el juego serio sea atractivo e inmersivo, el cual permite al usuario lograr adquirir las competencias objetivo. En fin, la metodología MPOBA combina tres aspectos: pedagógico, del juego y del software. Luego, se indica que el modelo es iterativo e inicia con la ingeniería de dominio.



**Figura 6.11** Modelo de proceso MPDSG

Fuente: Figura tomada del texto de (Evans et al., 2016, p. 9)

**Tabla 6.14 Resumen - Aspectos de análisis de MPDSG**

Aspectos generales	País Nivel educativo	Argentina Secundaria
Aspectos de diseño	Fundamentación	LPS, DCU, MPOBA
	Requerimientos	Entrevistas a docentes y reuniones con expertos en contenido o dominio.
	Definición de roles	Expertos en jugabilidad, expertos en pedagogía (docentes) y expertos del contenido o dominio.
	Patrones de diseño	No
	Reutilización de software	Si
Aspectos metodológico-pedagógicos	Documentación	Sí
	Diseño de prototipos	Sí
	Destinatarios	Docentes
Aspectos metodológico-pedagógicos	Intención pedagógica	Competencias del Siglo XXI: conocer, comprender, analizar, sistematizar y evaluar.
	Objetivos pedagógicos/lúdicos	Sí
Aspectos de análisis	Validación de objetivos	Sí, En todas las etapas del desarrollo, y validación integral como OA. Grupos de usuarios.

	Evaluación de UX Resultados cuantificables	Sí No
--	---	----------

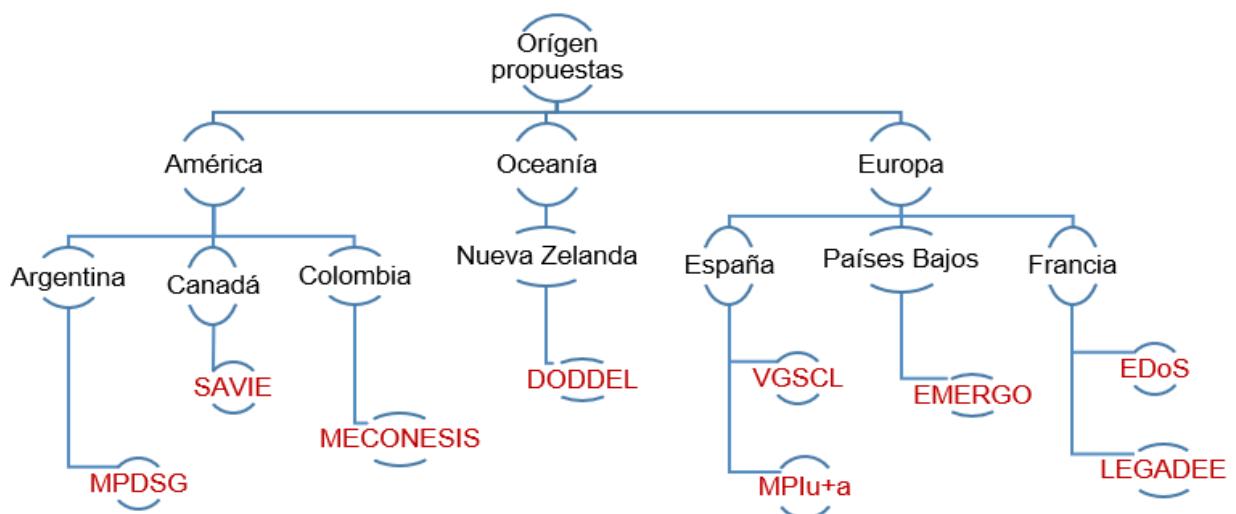
La metodología **MPDSG** se enfoca en garantizar la calidad del juego y el cumplimiento de los objetivos de jugabilidad e inmersión.

#### 4.12. Análisis de resultados según criterios de evaluación

Esta sección reúne los resultados obtenidos a partir de la aplicación de los criterios de análisis a las metodologías seleccionadas y descritas en la [Sección 6.6](#). Con el propósito de brindar una comprensión fluida del análisis, se brindan subsecciones acordes a las categorías de análisis, mismas que contiene los criterios definidos, mismos previamente determinados en el [Capítulo 5](#), se fundamentan según los aspectos teóricos descriptos y vinculados con el desarrollo de software.

##### 4.12.1. Aspectos generales

En relación con el **País donde se desarrolla la investigación**, se han analizado 9 propuestas diferentes de metodologías o procesos que buscan guiar el diseño de juegos serios. Tal como se puede observar en la [Figura 6.12](#), estas investigaciones se desarrollaron mayoritariamente en el continente europeo donde se registran 5 casos, 3 en el continente americano y 1 caso en Oceanía. Con relación al continente americano, únicamente se registra un caso en Colombia, Argentina y Canadá respectivamente. Siendo España y Francia los países que registran mayor número de propuestas metodológicas (2 cada uno).

**Figura 6.12 Criterio - País de origen según metodologías analizadas**

De acuerdo con la aplicación del criterio **nivel educativo**, la Tabla 6.15, permite observar los resultados obtenidos a partir de la aplicación de dicho criterio, el cual se enfoca en identificar los niveles educativos para los cuales se han orientado o pensado las metodologías revisadas. Además, permite analizar si una metodología ha sido pensada para ser utilizada para un grupo de usuarios en específico o para distintos grupos de poblaciones.

**Tabla 6.15 Criterio - Nivel educativo según metodologías analizadas**

Metodología	Nivel educativo					
	Especial	Inicial	Primario	Secundario	Terciario	Universitario
EMERGO						✓
EDoS						✓
LEGADEE						✓
SAVIE		✓	✓	✓	✓	✓
DODDEL					✓	✓
VGSC		✓	✓	✓		
MECONESIS	✓					
MPlu+a				✓		
MPDSG				✓		

El nivel educativo que registra mayor número de las metodologías analizadas corresponde a la educación superior/universitario con 5 de las 9 propuestas y, en segunda instancia el nivel secundario 4 metodologías. Además, SAVIE y VGSC son las únicas 2 metodologías que han sido pensadas o diseñadas para ser utilizadas en diferentes niveles educativos.

#### 4.12.2. Aspectos de diseño

El criterio de **fundamentación** permite conocer cuál ha sido el referente que ha sido tomado como base para el diseño de las propuestas metodológicas analizadas. Igualmente, permite examinar si la fundamentación se ha realizado a partir de uno o varios referentes. En la [Tabla 6.16](#), se muestra el análisis del criterio de fundamentación.

**Tabla 6.16 Criterio - Fundamentación según metodologías analizadas**

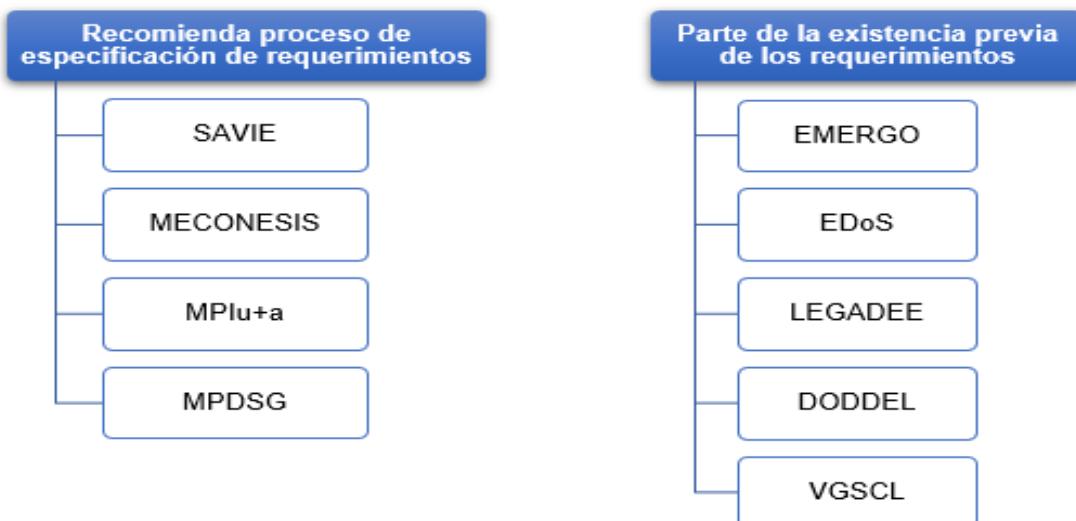
Metodología	Fundamentación de la metodología*					
	DCU	LPS	MPOBA	UP	ADDIE	LG
EMERGO					✓	
EDoS	✓					
LEGADEE		✓				
SAVIE	✓					
DODDEL					✓	✓
VGSCL	✓					
MECONESI	✓			✓		
MPLu+a	✓					
MPDSG	✓	✓	✓			

\***DCU:** Diseño Centrado en el Usuario. **LPS:** Líneas de Producción de Software. **MPOBA:** Modelo de Proceso para el Desarrollo de Objetos de Aprendizaje. **UP:** Proceso Unificado de Desarrollo de Software. **ADDIE:** Análisis, diseño, desarrollo, implementación y evaluación. **LG:** Metodología Centrada en Documentos.

En la [Tabla 6.16](#), se puede observar que, 6 de las nueve metodologías analizadas (EDoS, SAVIE, VGSCL, MECONESIS, MPLu+a y MPDSG) tomaron como referente para su fundamentación el Diseño Centrado en el Usuario (DCU). Igualmente, se puede constatar que, únicamente 3 de las propuestas (DODDEL, MECONESIS y MPDSG) utilizaron más de un referente para su fundamentación.

El criterio relacionado con la **definición de requerimientos** permite identificar si las metodologías revisadas consideran la fase o proceso de levantamiento de requerimientos del usuario dentro de su propuesta de diseño de juegos serios, o bien, se parte del hecho de que ya se conocen o se ha realizado previamente esta labor. En caso de que dentro de las metodologías se proponga esta fase, se podría realizar un registro de las recomendaciones acotadas para llevar adelante esta tarea. La [Figura 6.13](#), presenta las metodologías que han considerado dentro de su propuesta para el diseño de juegos serios, considerar la fase o proceso de especificación de requisitos por parte del usuario, misma que según diferentes autores (Prasad & Verma, 2016; Sommerville, 2016; Spinelli et al., 2017) debe

incluirse en el desarrollo de cualquier tipo de software, incluyendo los juegos serios, debido a que la definición de los requerimientos son concluyentes en relación con la calidad del software a desarrollarse (Alshazly et al., 2014).



**Figura 6.13 Criterio - Definición de requerimientos según metodologías analizadas**

Del total de las metodologías analizadas, únicamente 4 de ellas (SAVIE, MECONESIS, MPIu+a y MPDSG) consideran dentro de su propuesta de diseño de juegos serios, la inclusión de la fase o proceso de definición de requerimientos del usuario. Por su parte, las 5 metodologías restantes, inician su propuesta de diseño tomando como punto de partida el hecho de que ya se cuenta con un análisis de requerimiento efectuado, es decir, siquiera se hace alusión a esta tarea en las propuestas.

La metodología **MPDu+a**, lleva adelante la tarea de definición de requerimientos tomando como base la ingeniería de requisitos - ISO/IEC 9126-1. Luego, en **MPDSG** se indica que esta tarea se realiza a través de la implementación de entrevistas a docentes y reuniones con expertos en contenido o dominio, para tal tarea se sigue el proceso de elicidadón de requerimientos propuesto por (Spinelli et al., 2017). Seguido, **MECONESIS** realiza la definición de requerimientos a través de la aplicación de un cuestionario con una serie de preguntas cuidadosamente elaboradas para alcanzar tal propósito y, recomienda utilizar el modelo de tareas CTT para apoyarse en la obtención de los requerimientos tanto funcionales como no funcionales. Por último, **SAVIE** se recomienda tomar en cuenta la definición de

requerimientos por parte del usuario, sin embargo, no se identifica o se aborda la forma de realizarlo ni las herramientas para llevar adelante esta labor.

De acuerdo con (Cano, 2016; Marfisi-Schottman, 2012) la **definición de roles** es una etapa o fase de suma importancia en el momento de diseñarse un juego serio, debido a que en el proceso de diseño se debe contar con un equipo de trabajo multidisciplinario para poder alcanzar con éxito los objetivos pedagógicos y lúdicos. Este criterio permite identificar los roles que han sido definidos en las diferentes metodologías revisadas. En la [Tabla 6.17](#), se resumen los roles identificados que han sido definidos en cada una de las metodologías revisadas.

**Tabla 6.17 Criterio - Definición de roles según metodologías analizadas**

Metodología	Definición de roles				
	Expertos pedagógicos (Docentes)	Expertos de contenidos o dominios	Diseñadores y desarrolladores	Usuarios finales	Psicólogos
EMERGO	✓		✓	✓	
EDoS			✓		
LEGADEE	✓	✓	✓	✓	
SAVIE					
DODDEL			✓		
VGSCL	✓			✓	
MECONESI S	✓	✓		✓	✓
				✓	✓
MPlu+a					
MPDSG	✓	✓			

A nivel general, las metodologías estudiadas coinciden mayoritariamente en la definición de 3 roles: Expertos pedagógicos, usuarios finales, diseñadores y desarrolladores. Luego, en menor proporción, se coincide en el rol de expertos en contenidos o dominios y en el rol de psicólogos. Además, LEGADEE es la única metodología que incorpora una amplia variedad de roles (guionista, jefe de proyectos, director artístico, entre otros). En esta misma línea, la metodología MPDSG añade el rol de experto en jugabilidad. Por último, en la literatura revisada no se identifica claramente una definición de roles por parte de la metodología SAVIE, sin embargo, está orientada exclusivamente para el uso del profesorado.

El criterio de definición de **patrones de diseño** permite indagar si en las metodologías revisadas se considera dentro de su propuesta la definición de patrones de diseño. Acorde a lo expresado por (Cano, 2016; Sommerville, 2016), los

patrones de diseño permiten resolver problemas específicos a nivel del diseño, lo cual favorece la flexibilidad y reusabilidad del juego.

La Figura 6.14, permite observar que únicamente 4 de las 9 metodologías analizadas (MECONESIS, MPDSG, VGSCL y LEGADEE) consideran el uso de patrones de diseño dentro de su propuesta metodológica.

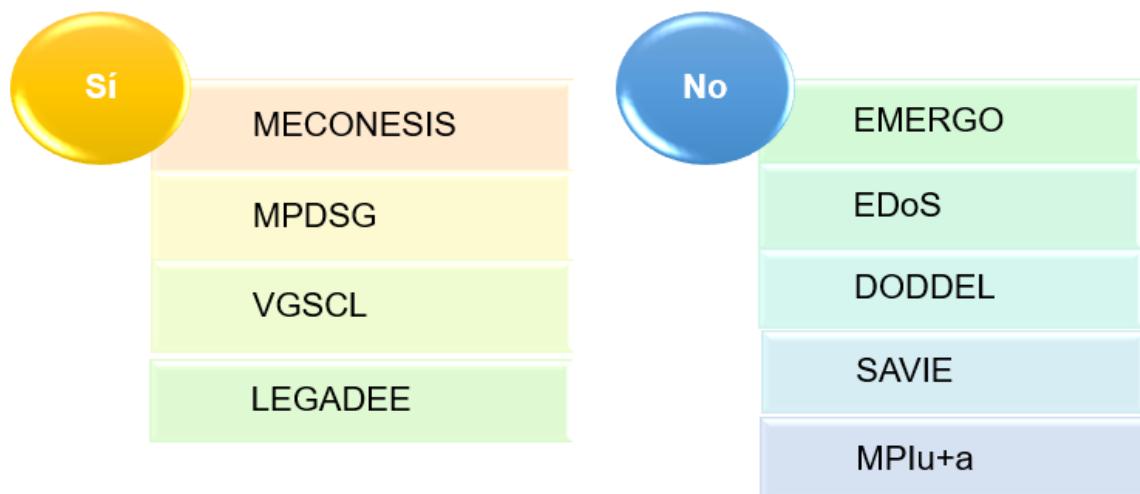


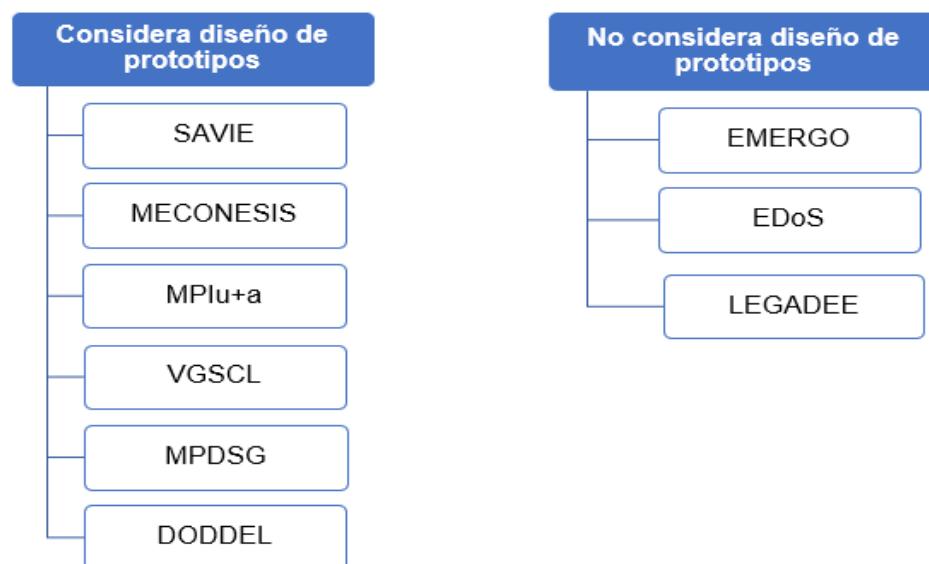
Figura 6.14 Criterio - Patrones de diseño

Además, es importante identificar si en las metodologías revisadas se considera la **reutilización de software**, debido a que permite flexibilidad para reutilizar contenidos o funcionalidades en diferentes instancias en un mismo proyecto, o bien, en otros proyectos (Ralph & Wand, 2007; Rani, 2017; Sommerville, 2016). Al aplicarse este criterio, se obtiene como resultado que 5 de 9 de las metodologías han considerado la reutilización de software, tal como se puede observar a detalle en la Figura 6.15. Igualmente, se puede apreciar que EDoS no indica el uso de patrones de diseño, sin embargo, sí considera la reutilización de software.

<input checked="" type="checkbox"/> Considera reutilización	<input type="checkbox"/> No considera reutilización
<input type="checkbox"/> MECONESIS	<input type="checkbox"/> EMERGO
<input type="checkbox"/> MPDSG	<input type="checkbox"/> DODDEL
<input type="checkbox"/> VGSCL	<input type="checkbox"/> SAVIE
<input type="checkbox"/> EDoS	<input type="checkbox"/> MPPlu+a
<input type="checkbox"/> LEGADEE	

**Figura 6.15 Criterio - Reutilización de software**

En relación con la información obtenida con respecto al criterio **diseño de prototipos**, se resalta como resultado que 7 de las 9 metodologías revisadas consideran recomendable el diseño de prototipos del juego por desarrollarse ([Figura 6.16](#)). Acorde con (Isaias & Issa, 2015; Lozano, 2017) los prototipos facilitan la identificación de errores y las posibles mejoras a realizar durante diferentes fases del diseño de software, lo cual podría garantizar la viabilidad del diseño (Buendía-García et al., 2013).

**Figura 6.16 Criterio – Diseño de prototipos**

El uso de prototipos usualmente es característico de las metodologías basadas en el DCU, debido a que el usuario final tiene un rol relevante durante el desarrollo del software (Cano, 2016; Evans et al., 2016), sin embargo, se podrían utilizar en el

desarrollo de software tradicional, tal como es el caso de DODDEL que incorpora la fase de creación de prototipos (rasgo innovador que la caracteriza).

#### 4.12.3. Aspectos metodológico-pedagógicos

Al analizar el criterio **destinatarios**, se observa en la [Tabla 6.18](#) que la mayor parte de las metodologías revisadas han sido pensadas para ser utilizadas por parte del profesorado a nivel de educación superior (7 de 9) para potenciar diferentes habilidades o competencias en el estudiantado. Además, se observa que algunas de las metodologías (EMERGO, SAVIE y MECONESIS) señalan como destinatario final al profesorado y estudiantado (viendo al estudiantado desde el punto de vista de la jugabilidad). Luego, 4 de las metodologías (LEGADEE, DODDEL, VGSC y MPi+u) se orientan a servir de guía para el equipo técnico en el diseño de juegos serios. Además, LEGADEE y VGSC permiten la interacción con el profesorado (el profesorado viéndolo desde el sentido de que, una vez diseñado el juego, este puede configurar ciertas actividades para utilizarlas con el estudiantado).

**Tabla 6.18 Criterio - Destinatarios**

<b>Metodología</b>	<b>Destinatarios</b>		
	<b>Estudiantes</b>	<b>Docentes</b>	<b>Equipo técnico*</b>
EMERGO	✓	✓	
EDoS		✓	
LEGADEE		✓	✓
SAVIE	✓	✓	
DODDEL			✓
VGSC		✓	✓
MECONESI	✓	✓	
MPi+u			✓
MPDSG		✓	

\*Equipo técnico: Diseñadores y programadores

Con base en el criterio de la **intención pedagógica**, el resultado más importante tiene que ver con que el total de las metodologías indagadas están orientadas a potenciar el aprendizaje, así como la formación de competencias o habilidades específicas: cognitivas, profesionales, genéricas, entre otras.

Igualmente, al analizar el criterio de **objetivos pedagógicos/lúdicos**, se obtuvo como resultado que, el total de las metodologías analizadas concuerdan en

que se debe incluir este aspecto dentro de las propuestas de diseño. Los objetivos claramente definidos favorecen la determinación de las competencias y los aprendizajes que se desean alcanzar o potenciar en el usuario final, así como la respectiva verificación del cumplimiento de los mismos (Cano et al., 2016).

#### 4.12.4. Aspectos de análisis

Para una mejor comprensión y con el propósito de ofrecer una mirada general, se ha agrupado en la [Tabla 6.19](#) los tres criterios vinculados con la evaluación y validación del proceso de diseño correspondientes a la categoría de **aspectos de análisis**. Específicamente en el criterio de **validación de objetivos**, se puede observar que las metodologías EDoS y DODDEL son las únicas en las que no se ha considerado la validación de objetivos. Luego, se observa que EMERGO, EDoS y LEGADEE tampoco realizan una evaluación de las **experiencias del usuario**. Además, se identifica que VGSC y MECONESIS son las únicas metodologías que consideran la obtención de **resultados cuantificables**. Además, se evidencia que EDoS no considera algún tipo de validación o evaluación.

**Tabla 6.19 Criterios - Validación de objetivos, evaluación UX y resultados cuantificables**

Metodología	Aspectos de análisis		
	Validación de objetivos	Evaluación UX	Resultados cuantificables
EMERGO	✓		
EDoS			
LEGADEE	✓		
SAVIE	✓	✓	
DODDEL		✓	
VGSC	✓	✓	✓
MECONESI	✓	✓	✓
MPlu+a	✓	✓	
MPDSG	✓	✓	

En relación con la información resumida en la [Tabla 6.19](#), es importante indicar que la **validación de objetivos** es realizada por la metodología EMERGO a través de pruebas de entradas de datos y aplicación de cuestionarios. Luego, VGSC ejecuta monitoreos para validar el proceso de aprendizaje, mientras tanto, las metodologías MPlu+a, MECONESIS y MPDSG efectúan la validación durante todas

las etapas del desarrollo a través de modelos propios definidos para alcanzar tal propósito.

Luego, la **validación de las experiencias de los usuarios (UX)** se realiza en SAVIE y DODDEL a través de prototipos. En el caso de MPi+u es mediante la aplicación de pruebas y el feedback. En la metodología MECONESIS se utiliza un modelo de análisis y, finalmente, en VGSC se ejecuta la validación UX mediante la aplicación de entrevistas y cuestionarios.

Con relación a la **medición de resultados**, la metodología MECONESIS emplea una etapa propia diseñada para tal propósito, mediante la cual se obtienen diferentes resultados (cuantitativos y cualitativos). Luego, VGSC realiza la medición a través de la aplicación directa de entrevistas y cuestionarios.

#### 4.13. Conclusiones del capítulo

En este capítulo se realizó un abordaje y descripción de diferentes metodologías desarrolladas para orientar el diseño de juegos serios, mismas que se analizaron de acuerdo con una serie de criterios que fueron definidos previamente.

A nivel de la categoría de **aspectos generales**, se logra identificar que el desarrollo de propuestas metodológicas se ha concentrado en el continente europeo con un 56% de las metodologías analizadas, seguido por el continente americano con el 33% y, por último, el asiático con el 11%. Con relación al nivel educativo, se enfocan mayormente para ser utilizadas en educación superior/universitario (56%).

En relación con los **aspectos de diseño**, se evidencia que, por las características propias de los juegos serios (pedagógicas y lúdicas) el usuario final adquiere un rol de importancia durante el diseño del juego, por ello, en la mayoría de las propuestas (67%) se consideró para su fundamentación, el diseño centrado en el usuario. Además, en el 56% de las metodologías se partió del supuesto de la existencia previa de la definición de requerimientos por parte del usuario. Sin embargo, el otro 44% metodologías consideran que es de vital importancia, en la fase inicial del diseño de una metodología, el incorporar los procesos de elicitation y especificación de requerimientos. Luego, la definición y asignación de roles de los diferentes actores que intervienen en la producción del juego fue considerada de importancia por parte del 56% de las metodologías, destacándose el rol docente (experto pedagógico), igualmente el 56% considera de importancia la inclusión del

rol de usuario final. Se resalta que estos procesos se deben realizar de forma conjunta y consensuada entre los diferentes actores involucrados. Igualmente, la flexibilidad y reusabilidad de componentes (reutilización de software) resultó de interés para el diseño de un juego. Además, se concluye que el uso de prototipos permite la reducción y corrección de posibles errores que puedan surgir durante el diseño del juego serio.

En cuanto a los aspectos **metodológico-pedagógicos**, se identifica que el 100% de las metodologías coinciden en que se debe realizar una definición previa de los destinatarios, en la cual el profesorado y el equipo técnico (programadores y desarrolladores) resultaron ser los más destacados. Asimismo, la definición de objetivos pedagógicos/lúdicos fueron considerados dentro de las propuestas del 100% de las metodologías, debido a que podrían permitir determinar competencias y aprendizajes a potenciar en el jugador. Además, se identifica que los objetivos pedagógicos están estrechamente relacionados con el contexto de uso y, por ello, se considera de importancia para el 100% de las metodologías, el que se definan en la fase inicial, para luego evaluar la calidad del juego en la fase final.

En relación con los **aspectos de análisis**, se identifica que el 78% de las propuestas metodológicas consideran de importancia la validación de objetivos y, el restante 22% no lo considera. En relación con la evaluación de las experiencias del usuario, el 67% de las metodologías considera importante su validación y el otro 33% no lo considera así. Aspectos que son validados a través del uso de prototipos, aplicación de pruebas y la utilización de diferentes tipos de *feedback*. Por ello, es de importancia el incorporar, en el diseño, la definición de elementos pedagógicos, ya que estos servirán para evaluar la eficiencia, el aprendizaje y la experiencia del usuario en relación con la interacción con el juego. Además, estos elementos pedagógicos permiten confeccionar un registro cuantitativo del aprendizaje, con el propósito de conocer el proceso y la calidad de este por parte del usuario.

Con base en los resultados anteriores, obtenidos durante el análisis de las metodologías estudiadas, se ha elaborado un **checklist** donde se resume algunas de las características distintivas o representativas identificadas que podrían ser consideradas al definir una metodología que permita guiar el diseño y desarrollo de juegos serios:

- ✓ Considerar el diseño centrado en el usuario.
- ✓ Incorporar la definición de requerimientos por parte de los usuarios.
- ✓ A nivel de asignación de roles, considerar el rol de experto pedagógico y del usuario final.
- ✓ Considerar la flexibilidad y reusabilidad de componentes (reutilización de software).
- ✓ Diseñar y crear prototipos del juego.
- ✓ Considerar como usuario final al profesorado a nivel de educación superior.
- ✓ Considerar a nivel de intención pedagógica potenciar habilidades y competencias específicas.
- ✓ Definir claramente los objetivos pedagógicos y lúdicos.
- ✓ Validar los objetivos durante la etapa del desarrollo.
- ✓ Evaluar la UX durante el diseño del juego serio.

En fin, se evidencian esfuerzos para consolidar metodologías orientadas al desarrollo de juegos serios, sin embargo, los resultados obtenidos permiten constatar que las metodologías analizadas se enfocan, a nivel general en brindar diferentes recomendaciones vinculadas con el diseño de un juego serio (Cano, 2016; Padilla-Zea, 2011). Es decir, en la revisión de literatura no se logra identificar concretamente una propuesta metodológica que sirva para orientar o guiar paso a paso el proceso para el desarrollo de juegos serios orientados a potenciar competencias tecnológicas. Sin embargo, se concluye que el conjunto de metodologías analizadas brinda una luz en términos de las consideraciones que se deben tener en cuenta al proponer este tipo de metodologías, máxime si se piensa a futuro desarrollar una propuesta metodológica para el diseño de juegos serios

orientados a potenciar competencias tecnológicas. Los cuales son el foco de investigación de esta tesis.

# **CAPÍTULO VII. REVISIÓN DE ARQUITECTURAS UTILIZADAS PARA EL DISEÑO DE VIDEOJUEGOS**

#### 4.14. Introducción

Este capítulo aborda y describe diferentes experiencias o investigaciones relacionadas con la implementación de arquitecturas que han sido propuestas para el diseño y desarrollo de videojuegos y, en particular, juegos serios. Las experiencias han sido seleccionadas con base en la revisión sistemática de la literatura existente y, de acuerdo con el punto de vista de diferentes autores (Calderón & Ruiz, 2015; Cano, 2016; Cano et al., 2016; Kitchenham, 2004, 2007; Kitchenham et al., 2009, 2010; Petersen et al., 2015), entre otros.

En el capítulo, se establecen categorías y criterios que permiten analizar las fuentes primarias localizadas en función de los objetivos de estudio propuestos. Se definen 4 categorías de análisis: 1- aspectos generales, 2- aspectos de arquitectónicos, 3- aspectos pedagógicos y 4-aspectos de análisis. Luego, se procede a describir a nivel general cada una de las experiencias o investigaciones seleccionadas, donde se aborda el objetivo principal de cada investigación, así como la respectiva caracterización de la propuesta arquitectónica ofrecida por cada investigador.

Para brindar contexto a las categorías y criterios de análisis, se describe cada uno de ellos. Posteriormente, se procede a la aplicación de los criterios para cada una de las experiencias identificadas, así como su respectivo análisis a partir de los resultados obtenidos. Por último, se presentan las respectivas conclusiones del capítulo vinculadas con los aspectos más relevantes identificados en las propuestas arquitectónicas para el diseño de videojuegos o juegos serios analizadas.

Este capítulo brinda respuesta a las preguntas de investigación; Plx12 y Plx13 establecidas para el estudio y referidas en el Capítulo 1:

Plx12 ¿Cuáles arquitecturas se han utilizado para el diseño de videojuegos o juegos serios?, ¿qué antecedentes existen en este sentido?

Plx13 ¿Qué aspectos arquitectónicos se consideran en la literatura para el diseño de videojuegos o juegos serios?

En el próximo apartado se aborda y describe cada uno de los criterios que servirán de guía para realizar el análisis de las experiencias o propuestas arquitectónicas de videojuegos o juegos serios seleccionadas para el estudio.

#### 4.15. Definición de los criterios de análisis

A partir de la revisión sistemáticas de literatura, se identificaron una serie de documentos primarios que describen la aplicación de diferentes tipos de propuestas vinculadas con arquitecturas para el diseño o desarrollo de videojuegos. Al igual que en el apartado correspondiente al análisis de las metodologías diseñadas para el desarrollo de juegos serios, se aclara que estas propuestas arquitectónicas varían de acuerdo con el autor o investigador que las analiza y del objetivo propio de cada investigación.

Para alcanzar el objetivo de investigación propuesto, se establecen criterios para orientar el análisis de un grupo de propuestas o experiencias identificadas previamente. Para la selección de los criterios, se tomaron como referencia los propuestos en las investigaciones realizadas por (Calderón & Ruiz, 2015; Cano, 2016; Cano et al., 2016; Kitchenham, 2004, 2007; Kitchenham et al., 2009, 2010; Petersen et al., 2015); estos criterios permiten identificar y describir propuestas arquitectónicas para el diseño de videojuegos, en particular, los juegos serios. Para describir y analizar las propuestas o experiencias arquitectónicas, se definieron 4 categorías con diferentes criterios de análisis cada una. En este sentido, se utiliza como referencia el formato de definición de criterios utilizado para el análisis de las metodologías relacionadas con el diseño de juegos serios, que ha sido presentado previamente en la [sección 6.2](#), con el propósito de unificar el análisis relacionado con las arquitecturas.

Para describir y analizar las investigaciones (fuentes primarias) relacionadas con las propuestas arquitectónicas para el desarrollo de videojuegos o juegos serios, se definen 4 categorías con diferentes criterios de análisis cada una:

**A. Aspectos generales** - Los criterios de esta categoría son afines a las propuestas arquitectónicas o experiencias por analizar. Los indicadores, permiten identificar el país e idioma de procedencia y el nivel educativo al que está dirigida la misma.

**B. Aspectos de arquitectónicos** - Los criterios de esta categoría están vinculados a analizar la forma en que se definen los aspectos arquitectónicos de las experiencias por analizar. Se pretende identificar si las propuestas han sido fundamentadas en alguna arquitectura en específico, así como determinar los

dispositivos para los cuales ha sido orientada la propuesta arquitectónica. Igualmente, determinar si la propuesta incluye la reutilización de componentes.

**C. Aspectos pedagógicos** - Los criterios presentes en esta categoría permiten conocer los destinatarios de la arquitectura, el propósito o intención de la arquitectura y, el nivel de complejidad o de uso de esta.

**D. Aspectos de análisis** - Esta categoría busca conocer el tipo de producto final a desarrollar con la arquitectura, así como identificar los posibles resultados que se han obtenido al implementarla o utilizarla. Finalmente, determinar el tipo de ayuda que se ofrece al usuario.

En la [Tabla 7.1](#), se puede observar la clasificación de los criterios en cada una de las categorías detalladas (aspectos generales, aspectos arquitectónicos, aspectos pedagógicos y aspectos de análisis).

*Tabla 7.1 Criterios definidos para el análisis de las propuestas arquitectónicas*

Categoría	Criterios
Aspectos generales	País donde desarrolla la propuesta arquitectónica Idioma de la propuesta Nivel educativo.
Aspectos arquitectónicos	Fundamentación Tipo de dispositivo al que está orientada la arquitectura Reutilización de componentes
Aspectos pedagógicos	Definición de destinatarios de la arquitectura. Propósito o intención de la arquitectura. Nivel de complejidad.
Aspectos de análisis	Tipo de producto a desarrollar (juego) Resultados de utilización obtenidos. Guías o ayudas disponibles.

De acuerdo con las categorías y criterios definidos en la [Tabla 7.1](#), se realiza una breve descripción de los criterios de análisis propuestos para la revisión y estudio de las iniciativas o propuestas arquitectónicas para el diseño de videojuegos o juegos serios, con el propósito de brindar contexto o significado a cada uno y posteriormente, hacer más fácil su comprensión.

#### A. Aspectos generales

En esta categoría se consideran tres criterios: País donde desarrolla la investigación o propuesta arquitectónica, el idioma y el nivel educativo:

- ✓ **País donde desarrolla la propuesta arquitectónica** (*origen de la propuesta*)
  - Este criterio hace referencia al país donde se desarrolla o propone la arquitectura, o bien, país donde se desarrolla la investigación.
- ✓ **Idioma de la propuesta** - Este criterio hace referencia a los posibles idiomas en los cuales se podría encontrar disponible la propuesta arquitectónica, o bien, la investigación.
- ✓ **Nivel educativo** - Este criterio permite identificar el nivel educativo al que está dirigida la propuesta arquitectónica. Puede ser a nivel de educación especial, primaria, secundaria o superior/universitaria (grado y postgrado).

## B. Aspectos arquitectónicos

Esta categoría reúne 3 criterios: fundamentación, tipo de dispositivo al que está orientada la arquitectura y reutilización de componentes.

- ✓ **Fundamentación** - Este criterio permite identificar el fundamento o base que los investigadores han tomado de referencia para el diseño de la arquitectura. Los posibles valores para este criterio son; SOA, MVC, SE, ARGILE, ArJuS, entre otras.
- ✓ **Tipo de dispositivo al que está orientada la arquitectura** - Este criterio permite identificar el tipo de dispositivo al que está orientada la arquitectura. Los posibles valores para este criterio son: aplicación móvil, aplicación de escritorio o aplicación web.
- ✓ **Reutilización de componentes** - Este criterio permite identificar si en las arquitecturas o investigaciones analizadas se considera la reutilización de componentes. Los posibles valores para este criterio son; **sí / no**.

## C. Aspectos pedagógicos

Esta categoría engloba 3 criterios: definición de destinatarios de la arquitectura, propósito o intención de la arquitectura y nivel de complejidad.

- ✓ **Definición de destinatarios de la arquitectura** - Este criterio hace referencia al público meta para el cual está orientada la arquitectura, es decir, la población en específico, tal como desarrolladores, programadores, docentes o estudiantes.
- ✓ **Propósito o intención de la arquitectura** - Este criterio hace referencia al objetivo que caracteriza a la arquitectura, es decir, la habilidad o competencia que se pretende potenciar. Los valores posibles por obtener se vinculan con la formación o adquisición de habilidades o competencias (competencias genéricas, digitales, ambientales, salud, educación, entre otras) que se desean alcanzar en algún área en específico (área del conocimiento para las cuales han sido diseñadas).
- ✓ **Nivel de complejidad** - Este criterio permite determinar el grado de complejidad de la propuesta arquitectónica. Es decir, si la arquitectura es amigable con usuarios no expertos en el área de tecnologías. Los posibles valores para este criterio son: **nivel bajo** (personas con conocimientos muy básicos en herramientas de ofimática o nulos), **nivel medio** (requiere que las personas usuarias posean conocimientos avanzados en uso y ampliación de diferentes herramientas tecnológicas y, **nivel alto** (requiere que las personas usuarias dominen lenguajes de programación).

#### D. Aspectos de análisis

En esta categoría se incluyen 3 criterios: tipo de producto a desarrollar, resultados obtenidos y guías o ayudas disponibles:

- ✓ **Tipo de producto a desarrollar** - Este criterio permite determinar el producto a desarrollar con la propuesta arquitectónica. Los posibles valores para este criterio son; videojuego común o juego serio, otro. En todos los casos se identifica el nombre del producto a desarrollar.

- ✓ **Resultados de utilización obtenidos** - Este criterio permite determinar si se identifican resultados académicos y científicos que respalden los hallazgos obtenidos a partir de la implementación de la propuesta arquitectónica.
- ✓ **Guías o ayudas disponibles** - Este criterio permite analizar si en las experiencias seleccionadas se identifica algún tipo de ayuda que puede brindar la arquitectura a los usuarios (manuales, tutoriales, guías de apoyo, preguntas frecuentes, sugerencias, entre otros).

#### **4.16. Recopilación de investigaciones relacionadas con propuestas de arquitecturas para el diseño o desarrollo de videojuegos**

A partir de la revisión sistemática de literatura, se identifica un conjunto de investigaciones o fuentes primarias vinculadas con propuestas arquitectónicas para el desarrollo de videojuegos o juegos serios, las cuales han sido seleccionadas para el análisis y se pueden observar en la [Tabla 7.2](#).

**Tabla 7.2 Referencias seleccionadas para el análisis**

<b>Autor/es</b>	<b>Nombre de la referencia o investigación</b>	<b>Fuente</b>
(Maggiorini et al., 2016)	SMASH: A Distributed Game Engine Architecture.	IEEE Xplore
(El Mawas, 2014)	An Architecture for Co-designing Participatory and Knowledge-intensive Serious Games.	IEEE Xplore
(Gutiérrez-Hernández et al., 2013)	Arquitectura de Software para Juegos Serios con Aspectos Culturales: Caso de Estudio en un Videojuego para Formulas Temperatura	Semantic Scholar
(Söbke & Streicher, 2016)	Serious Games Architectures and Engines.	Springer
(Stavrev et al., 2018)	Concepts for distributed input independent architecture for serious games.	ResearchGate
(Flores et al., 2019)	Arquitectura de un juego serio inteligente basado en retos de matemáticas básicas.	Semantic Scholar
(Mizutani et al., 2021)	Software architecture for digital game mechanics: A systematic literature review.	ELSEVIER
(Ollsson et al., 2015)	Evolution and Evaluation of the Model-View-Controller Architecture in Games.	IEEE Xplore
(Carvalho, Bellotti, Hu, et al., 2015)	Towards a Service-Oriented Architecture framework for educational serious games	IEEE Xplore

Autor/es	Nombre de la referencia o investigación	Fuente
(Scacchi, 2017)	Practices and Technologies in Computer Game Software Engineering.	IEEE Xplore
(Yessad et al., 2010)	SeGAE: Serious Game Authoring Environment.	IEEE Xplore
(Costa et al., 2016)	Architecture to Portals of Serious Games and Virtual Environments with Performance Evaluation during Sequences of Activities.	IEEE Xplore
(Ismail & Belkhouche, 2019)	A Reusable Software Architecture for Personalized Learning Systems.	IEEE Xplore

A continuación, se describe brevemente cada una de las metodologías citadas, las cuales han sido diseñadas en diferentes países.

#### 4.16.1. Propuesta de (Maggiorini et al., 2016)

Usualmente, para el diseño o desarrollo de videojuegos se requiere de la colaboración de expertos en diferentes áreas del conocimiento (programadores, diseñadores, entre otros) quienes ejecutan tareas específicas (Maggiorini et al., 2016). Con el propósito de realizar una mejor asignación de habilidades y de esfuerzos, así como propiciar la reutilización de código y recursos, se ha utilizado en la implementación de los videojuegos los *game engines* o motores de juego, los cuales generalmente están organizados como una pila (*Stack*) de software en el sistema operativo, con niveles de abstracción crecientes. Sin embargo, la adopción de esta arquitectura para el desarrollo de juegos de gran escala ha presentado ciertas desventajas, entre ellas; inconvenientes el crear videojuegos centralizados, monolíticos, difíciles de escalar y dependientes de una sola plataforma (Maggiorini et al., 2016).

En este sentido, los investigadores **italianos** (Maggiorini et al., 2016) proponen una arquitectura distribuida para motores de juego, la cual denominaron (SMASH, *Stackless Microkernel Architecture for Shared environments*) (Arquitectura de Microkernel sin Pila para Entornos Compartidos). SMASH, permite dividir un motor de juego en múltiples módulos de software dinámicos e independientes, que interactúan entre sí, a través de mensajes por bus, al estilo de *microkernel*.

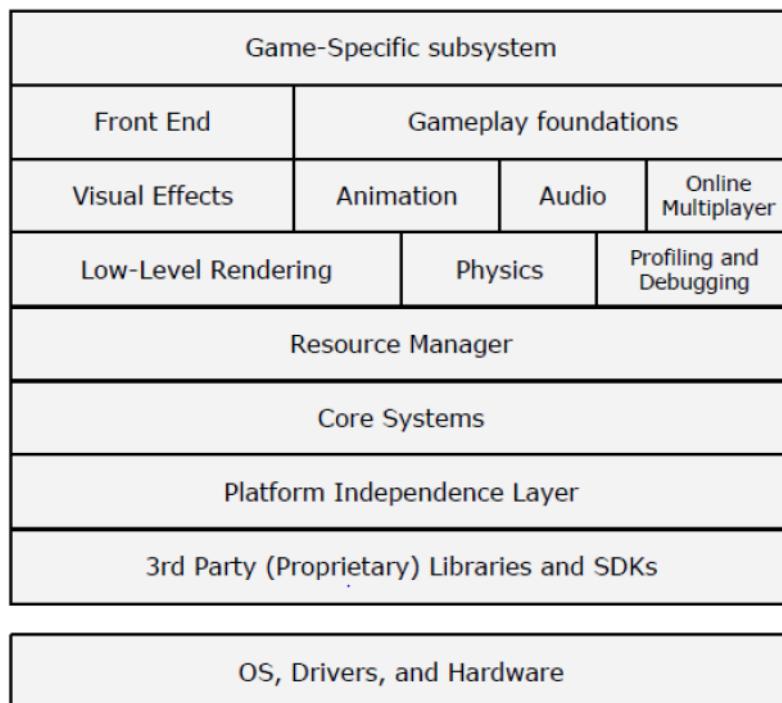
Los motores de juego permiten abstraer detalles y tareas en específicas en el proceso de creación de juegos, tales como el *rendering*, la física y el manejo de los

datos ingresados por el usuario; de esta manera, los desarrolladores pueden poner su atención mayoritariamente en implementar características específicas del juego (Maggiorini et al., 2016).

En general, el modelo arquitectónico empleado por la mayoría de motores de juego actuales presentan principalmente tres defectos (Maggiorini et al., 2016):

1. **Juegos monolíticos** - Todo el programa se debe recompilar a la hora de realizar cualquier cambio en el código.
2. **Juegos centralizados** - Todo el peso computacional recae sobre una misma máquina; la carga no se distribuye y puede afectar al rendimiento.
3. **Dependencia de plataformas** - Todo el código de los desarrolladores puede estar diversificado entre distintas plataformas.

En la [Figura 7.1](#) se muestra la arquitectura estándar para un motor de juego.



**Figura 7.1 Resumen de una arquitectura estándar para motor de juego**

Fuente: Figura tomada del texto de (Maggiorini et al., 2016, p. 2)

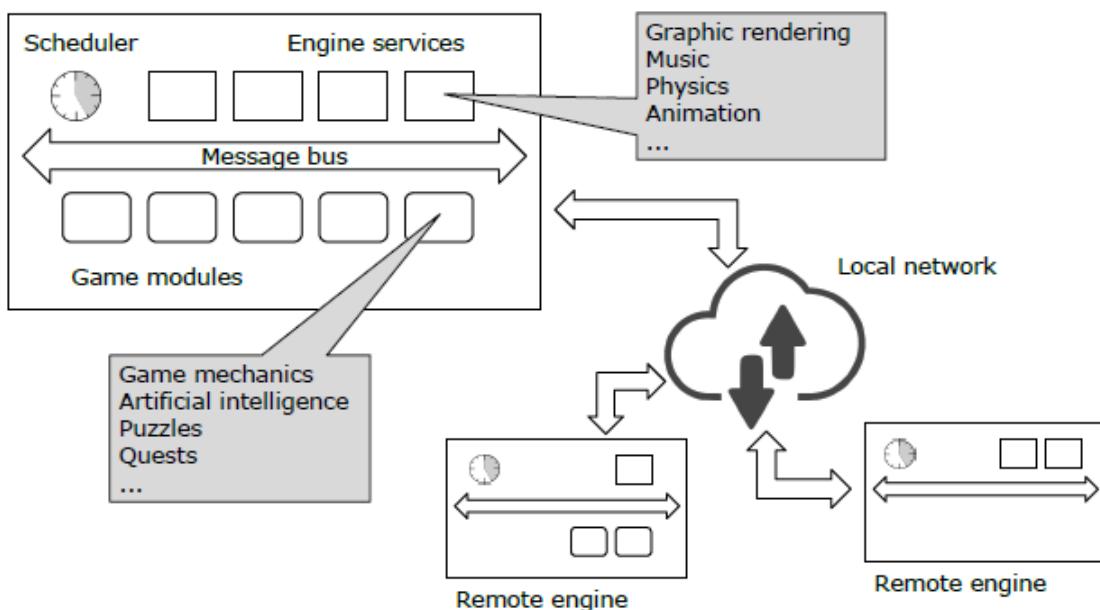
La arquitectura SMASH está inspirada –en sus bases más fundamentales– en el funcionamiento de un motor (metafóricamente). Posee una aproximación que permite acercar más el funcionamiento del motor de juego al de un entorno de ejecución (*runtime environment*), más que a un *stack* de bibliotecas (Maggiorini et al., 2016). Un motor de juego con arquitectura SMASH provee tres funcionalidades

básicas: *Soft real-time scheduler*, administrador dinámico de módulos de juego y un sistema de mensajería entre módulos (Maggiorini et al., 2016).

Entre los aspectos relevantes de la arquitectura se destacan los siguientes (Maggiorini et al., 2016):

- ✓ El *soft real-time scheduler* se encarga de cerciorarse de que el sistema está trabajando al ritmo adecuado.
- ✓ Los módulos de juego son entidades independientes que proveen funcionalidades.
- ✓ Los desarrolladores poseen la capacidad de modificar y extender los juegos vía *plug-ins*.
- ✓ Los juegos creados no serán monolíticos.
- ✓ Los errores de compilación de módulos en específico no representarán un retraso en las partes no involucradas.
- ✓ El bus de mensajería será el corazón de SMASH, ya que, la comunicación entre módulos es de extrema importancia.
- ✓ Cada módulo podrá invocar, mediante este bus de mensajería a diferentes funciones de otros módulos.
- ✓ Se podría crear instancias múltiples de los módulos, en diferentes motores.
- ✓ Es una arquitectura dinámica.

En la Figura 7.2 se puede observar el diagrama básico de la arquitectura propuesta e inspirada en arquitecturas de *microkernel*.



**Figura 7.2** Diagrama de la arquitectura SMASH

Fuente: Figura tomada del texto de (Maggiorini et al., 2016, p. 3)

Entre las posibles desventajas que podría presentar la arquitectura de la Figura 7.2 están: i) se requiere contar con la información sobre la topología completamente correcta y clara en cada nodo, ii) se debe lograr un balanceo óptimo de cargas para su correcto funcionamiento, iii) los buses de mensajería deben reaccionar inmediatamente ante cualquier cambio (Maggiorini et al., 2016).

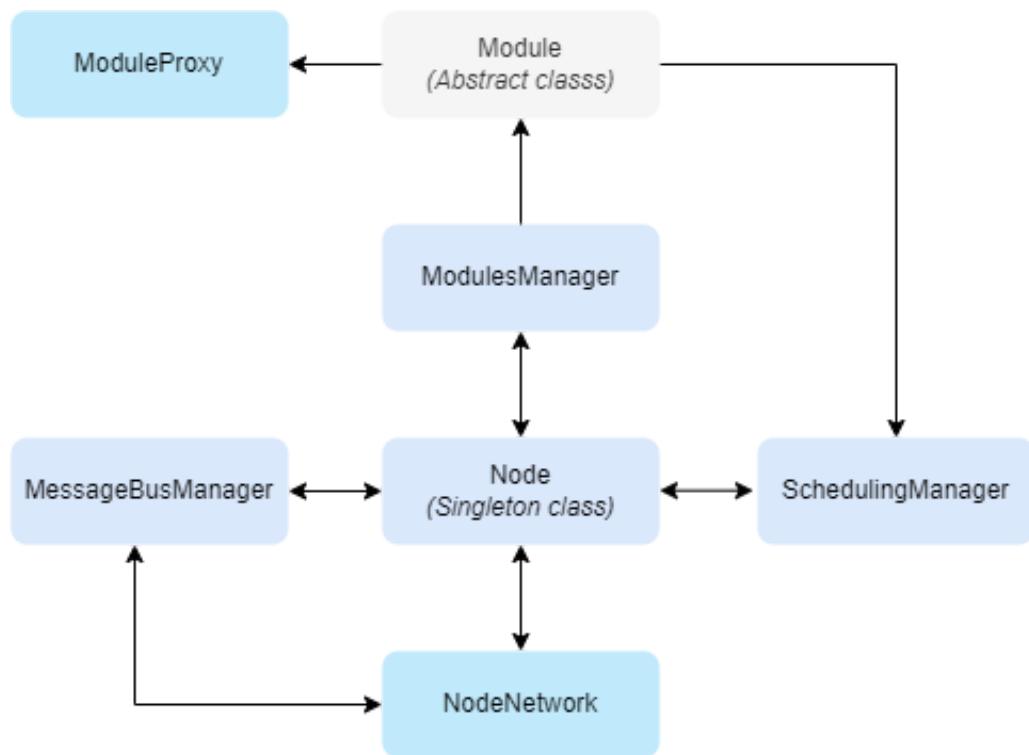
Al implementarse dicha arquitectura, se debe considerar que el lenguaje de referencia elegido puede afectar el nivel de adopción que tendrá el motor de juego. Por ello, los investigadores (Maggiorini et al., 2016) eligieron para este caso en particular, el lenguaje C#, debido a que posee capacidades de reflexión y cargado dinámico de clases, facilita una curva de aprendizaje gradual (no abrupta) y, puede interoperar con otros lenguajes de programación orientados a objetos.

Se realizó un análisis de desempeño en ambiente de ejecución para asegurar que este puede satisfacer los requerimientos de un juego al trabajar con un proxy en el bus de mensajería que media las llamadas de los módulos. Para alcanzar tal propósito, se realizaron test exhaustivos con una aplicación de prueba, resultando que:

- ✓ El tiempo de ejecución normal fue de 7 nanosegundos.

- ✓ Al incluir el proxy, este tiempo aumentó 27 veces.
- ✓ Al agregar etapas de red (TCP/UPD), el tiempo aumentó 10.000 veces.

Los resultados permitieron determinar que el sistema puede funcionar con 15.000 interacciones sincrónicas por segundo, por lo tanto, el desempeño en esta arquitectura resultó satisfactorio acorde a los requerimientos previos definidos (Maggiorini et al., 2016). En la Figura 7.3 se muestra el diagrama de clase de un prototipo en SMASH.



**Figura 7.3** Diagrama de la arquitectura SMASH  
 Fuente: Elaboración propia a partir del texto de (Maggiorini et al., 2016, p. 5)

Como se puede observar en la Figura 7.3, existe una clase para cada función básica del motor:

- ✓ **Node** - Administración del motor.
- ✓ **Module** - Clase abstracta para la creación de módulos.

- ✓ **ModulesManager** - Carga y descarga de módulos en el motor.
- ✓ **SchedulingManager** - Posee un temporizador de alta resolución y planifica temporalmente las peticiones que vienen de los módulos.
- ✓ **MessageBusManager** - Administra el bus de mensajería, que lleva las llamadas entre módulos.
- ✓ **ModuleProxy** - Intermediario entre módulos y funciones pertenecientes a otros módulos.
- ✓ **NodeNetwork** - Administra las conexiones con otros motores, y sus módulos y funciones.

Para probar la funcionalidad de la arquitectura, se implementó un juego de rompecabezas: un simulador de cubo de Rubik. En este juego se utilizaron tres computadoras conectadas en Full Mesh, cada una corriendo una copia independiente del prototipo. Cada computadora poseía módulos diferentes encargados de distintas funciones: una vela por la simulación del modelo, otra por los inputs y el renderizado en 2D, y otra por el renderizado en 3D del cubo. Durante las pruebas de rendimiento se obtuvo como resultado que el sistema es apto para albergar una aplicación de tiempo real, tal como los videojuegos o juegos serios (Maggiorini et al., 2016).

**Tabla 7.3 Resumen - Aspectos de análisis de (Maggiorini et al., 2016).**

Categoría	Criterios	Descripción
Aspectos generales	País de la propuesta arquitectónica. Idioma de la propuesta. Nivel educativo.	Italia. Inglés. Educación Superior.
Aspectos arquitectónicos	Fundamentación. Tipo de dispositivos. Reutilización de componentes	Microkernel. Móvil, escritorio y web. Sí.
Aspectos pedagógicos	Destinatarios de la arquitectura. Propósito de la arquitectura. Nivel de complejidad.	Desarrolladores. Creación de motores de juego. Alto.
Aspectos de análisis	Tipo de producto a desarrollar (juego)  Resultados de utilización obtenidos.	Motores de juego para videojuegos comunes y serios. No se evidencian resultados concretos en la literatura.

Categoría	Criterios	Descripción
	Guías o ayudas disponibles.	Descripción técnica de sus funcionalidades.

#### 4.16.2. Propuesta de (El Mawas, 2014)

En **Francia**, los investigadores (El Mawas, 2014) presentan por primera vez la arquitectura para representaciones, juegos, interacciones y aprendizaje entre expertos (ARGILE, *Architecture for Representations, Games, Interactions, and Learning among Experts*), la cual ha sido considerada por los propios investigadores como apropiada para juegos serios participativos e intensivos en la adquisición de habilidades y conocimiento.

El propósito de la investigación consiste en ofrecer una arquitectura para el diseño de juegos serios que permitan potenciar la adquisición o formación de competencias en prácticas de gestión de crisis, toma de decisiones en sistemas sociotécnicos complejos, donde el conocimiento transmitido se encuentra en evolución continua. Las reglas y los objetivos del juego son directamente generados de un foro especial de discusión. En este sentido, El Mawas (2014) propone un conjunto de criterios, metodologías y experimentos para verificar que las reglas y objetos del juego sean fácilmente anotados, discutidos y modificados por entrenadores y expertos del área mediante el uso de ARGILE.

Según (El Mawas, 2014), las reglas y elementos de los juegos serios son el resultado de un diseño participativo y descentralizado. Por un lado, es un diseño centrado en el usuario y, por otro lado, se confrontan profesores, diseñadores, actores del dominio, compañeros locales, expertos o científicos de diversas disciplinas. Las reglas y objetos pueden ser constantemente comentadas, discutidas y modificadas por formadores y actores del dominio, sin necesidad de un intermediario de especialistas en Tecnologías de Información (TI). El contexto del juego diseñado en esta arquitectura, es un juego multijugador que permite la formación de competencias para el trabajo en equipo (El Mawas, 2014).

Para la construcción de la arquitectura, se toman en cuenta varios modelos, enfoques y metodologías de diseño como el enfoque centrado en el uso de una herramienta técnica propuesto por (Robertson & Nicholson, 2007), el “modelo centrado en contenido” de (Moreno-Ger et al., 2008), el “Modelo DODDEL” de (McMahon, 2009), entre otros. Con ellos (El Mawas, 2014), propone una

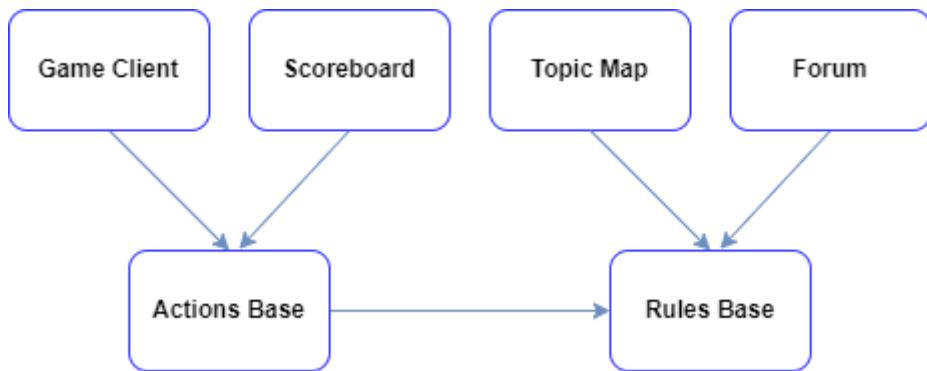
clasificación donde se toma en cuenta la simplicidad en términos del uso del dispositivo, la colaboración entre diseñadores, guía para los diseñadores en la fase de diseño y el involucramiento de expertos en la fase de diseño. Dicha clasificación se muestra en la [Tabla 7.4](#).

**Tabla 7.4 Comparativa de diferentes metodologías de diseño para juegos serios**

	Principios	Simplicidad	Colaboración entre diseñadores	Guía de diseño	Expertos en la fase de diseño
(Robertson & Nicholson, 2007)	Enfoque centrado en el uso de una herramienta técnica	x	x	x	
(Moreno-Ger et al., 2009)	El modelo centrado en el contenido				x
(McMahon, 2009)	Modelo DODDEL	x	x	x	
(Huynh-Kim-Bang et al., 2010; Björk et al., 2004)	Patrones de diseño para juegos serios			x	
(Sauvé, 2010)	Metodología de los juegos de marco genérico	x			x
(Harteveld, 2011)	El modelo de diseño de juegos triádicos			x	
(Marfisi-Schottman, 2012)	El proceso de ingeniería de los juegos serios	x	x	x	x

**Fuente:** Tabla adaptada a partir del texto de (El Marwa, 2014, p. 390)

A partir de los datos obtenidos en la [Tabla 7.4](#), se llegó a la conclusión de que “El proceso ingenieril de los juegos serios” propuesto por (Marfisi-Schottman, 2012) es la única metodología de diseño que cumplía con todos los aspectos propuestos por (El Mawas, 2014), no obstante, no se tomó en cuenta porque no cumplía un requisito clave, el cual era el del intercambio de puntos de vista en las diferentes etapas de diseño de un juego serio participativo e intensivo en conocimiento. Por lo que se propuso una arquitectura técnica ([Figura 7.4](#)), en la cual para ser editada por personas no-especialistas en TI, las reglas del juego deben ser tratadas como datos y no como programas.

**Figura 7.4 Dependencia entre los componentes de la arquitectura**

Fuente: Elaboración propia a partir del texto de (El Mawas, 2014, p. 393)

La investigación realizada por (El Mawas, 2014), se realizó de forma conjunta con el grupo SOS-21, los cuales desarrollaron un juego de aventura titulado “*point-and-click*”, el cual se ajusta a la dependencia de los componentes mostrada en la Figura 7.4. Se consideró que las reglas del juego son un punto de referencia para el mapa de temas, el foro y el cálculo de la puntuación, por ello, la importancia de ser un servicio compartido. De forma más clásica, otro servicio permite la gestión de las acciones de los jugadores. Este servicio no solo permite el conocimiento mutuo entre jugadores, sino que también permite a los diseñadores acceder a un marcador para el análisis de los rastros de uso (El Mawas, 2014).

Para la evaluación preliminar se utilizó el foro de ARGILE por dos semanas con el fin de definir reglas de juego para un juego participativo e intensivo en conocimiento que desea enseñar a los jugadores, buenas prácticas en el desarrollo sostenible. Esta experiencia se dividió en cuatro etapas:

- ✓ **Preparación para la fase de diseño** - Los diseñadores fueron 13 estudiantes y 5 investigadores del departamento IMESD (*Ingénierie et Management de l'Environnement et du Développement Durable*) de la UTT (*Université de Technologie de Troyes*). Los diseñadores utilizaron el foro de ARGILE para especificar y definir las reglas del juego. Dichas reglas deben de tomar en cuenta el gasto de energía, huella del CO<sub>2</sub>, gasto económico, dimensión ecológica, entre otros. La misión de los diseñadores fue de tomar en cuenta estas reglas para amueblar una casa vacía sin calefacción, decoración ni muebles.

- ✓ **Diseño del juego** - Se creó un directorio principal con la escena del juego para permitir a los diseñadores votar, agregar y discutir objetos, subescenas y acciones. Para definir diferentes temas de discusión, se dividió al grupo en 6, cada grupo puede interactuar con los otros y brindar diversos puntos de vista.
- ✓ **Encuesta de diseñadores** - Se realizaron encuestas a los diseñadores sobre temas como el uso práctico del foro ARGILE (acceso, equipo), el uso de características del foro y la integración del foro ARGILE en la fase de diseño de un juego serio participativo e intensivo en conocimiento.
- ✓ **Análisis de la encuesta** - Al analizar las respuestas, se notó que a los diseñadores no se les dificultó el ingreso ni el uso de la plataforma. También le pareció bueno los aspectos de colaboración y el trabajo a distancia. Algunos grupos empezaron con una estructura de árbol muy complicada, no obstante, al observar otros grupos que ordenaron los objetos de otra manera, lograron simplificar sus árboles. Además, esta experiencia mostró que dicha plataforma facilita los escenarios de diseño de juegos serios participativos e intensivos en conocimiento, ya que, los diseñadores fueron exitosos en el diseño de escenarios de juego para expresar y discutir conocimiento complejo, debido a que en las dos semanas de prueba se crearon 36 objetos.

En conclusión, la arquitectura ARGILE, permite a los expertos en áreas que no son de TI, ser diseñadores de juegos. Dicha plataforma apoya diferentes puntos de vista y debates entre expertos. También ofrece un enfoque constructivista hacia el aprendizaje por medio del foro ARGILE y se puede lidiar con el conocimiento que está en constante evolución. Por otro lado, se propone una arquitectura técnica y flexible para lidiar con las reglas del juego en forma de datos y no como programas, para que los expertos puedan modificar o agregar un escenario sin necesidad de la intervención de un especialista en TI. Además, se valida la hipótesis de que la discusión a través del foro hace que el estudiantado domine por sí mismos el aprender y promover la transición del modelo transmisivo de conocimiento al modelo colaborativo de comunidades de aprendizaje (El Mawas, 2014).

Tabla 7.5 Resumen - Aspectos de análisis de (*El Mawas, 2014*)

Categoría	Criterios	Descripción
Aspectos generales	País Idioma de la propuesta Nivel educativo	Francia. Inglés. Educación Superior.
Aspectos arquitectónicos	Fundamentación Tipo de dispositivos Reutilización de componentes	ARGILE. Aplicación web. No.
Aspectos pedagógicos	Destinatarios de la arquitectura Propósito de la arquitectura Nivel de complejidad	Docentes y estudiantes. Formación de competencias. Nivel medio.
Aspectos de análisis	Tipo de producto a desarrollar (juego) Resultados de utilización obtenidos Guías o ayudas disponibles	Juego Serio sobre buenas prácticas en el desarrollo sostenible. Resultados positivos sobre el foro ARGILE. No.

#### 4.16.3. Propuesta de (Gutiérrez-Hernández et al., 2013)

En México, los investigadores (Gutiérrez-Hernández et al., 2013) propusieron una arquitectura que integra aspectos culturales para un juego serio multicultural. La arquitectura se diseñó pensando en juego serios que posean diversas actividades lúdicas y de aprendizaje. Se utilizó la arquitectura para desarrollar un juego serio llamado "Fórmulas Temperatura", el cual permitió capturar diferentes mediciones entre países y regiones (Gutiérrez-Hernández et al., 2013).

En esta línea, los investigadores (Gutiérrez-Hernández et al., 2013), identificaron algunas problemáticas que se podrían presentar al crear juegos serios culturales, entre ellas:

- ✓ **Invisibilidad** - Se dificulta lograr conseguir la invisibilidad de las características de los aspectos culturales de los juegos.
- ✓ **Costo** - el costo de desarrollo podría ser elevado.
- ✓ **Conocimientos amplios** - Este tipo de juegos serios están diseñados solo por diseñadores de juegos: pueden ser muy entretenidos, pero la adquisición de conocimientos es amplia.

- ✓ **Motivación y compromiso** - El profesorado puede diseñar juegos serios, los cuales pueden resultar educativamente eficientes, sin embargo, pueden carecer de estrategias para motivar y comprometer al jugador.
- ✓ **Adaptación de guiones gráficos** - Se dificulta la adaptación de los *storyboards* (guiones gráficos) al profesorado, al adaptar los niveles del videojuego a los objetivos pedagógicos cuando se realiza el desarrollo del juego.
- ✓ **Aptitud** - Las aptitudes del personal involucrado en el desarrollo del juego, no suelen aprovecharse en problemas adecuados y en el tiempo correcto.

Por consiguiente, los investigadores (Gutiérrez-Hernández et al., 2013) propusieron un modelo arquitectónico que presente aspectos culturales con el fin de facilitar el diseño, producción de juegos serios educativos, para identificar cómo se reflejan las características culturales de la arquitectura de software y qué personal se involucra en el proceso de desarrollo.

La arquitectura propuesta se basó en seis facetas (objetivos pedagógicos, dominio simulación, interacciones con la simulación, problemas y progresión), para ello, se consideraron los aspectos culturales de la experiencia pedagógica y la de diseño de videojuego. La Tabla 7.6 permite mostrar dicha clasificación:

**Tabla 7.6 Facetas de diseño del juego serio**

No.	Faceta	Experiencia Pedagógica	Experiencia Diseño Videojuego
1	Objetivos Pedagógicos	X	
2	Dominio Simulación	X	
3	Interacciones con las Simulación		X
4	Problemas y Progresión	X	X
5	Decoración	X	
6	Condiciones de Uso	X	

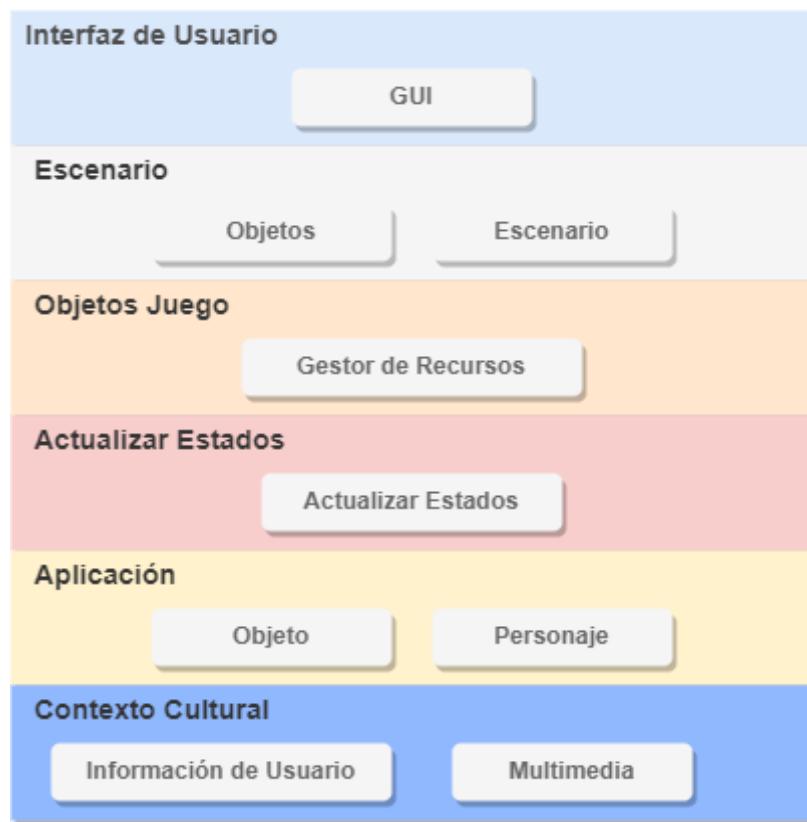
**Fuente:** Tabla adaptada a partir del texto de (Gutiérrez-Hernández et al., 2013, p.2)

Acorde con la Tabla 7.6, en la fase de **objetivos pedagógicos** se identifican las competencias, además, se define el contenido pedagógico del juego y los

objetivos educativos. En el **dominio simulación** se simulan casos específicos y se construyen modelos de la interfaz gráfica. En la fase de **interacciones con la simulación** se da la interacción pedagógica-social (los indicadores avanzados, la validación de competencias externas, preguntas, nuevas perspectivas y respuestas). Luego, en la fase de **problemas y progresión**, se refiere a los logros y cómo estos son medidos; es decir, el método de ludificación utilizado (curva de aprendizaje suave, sorpresas, puntuaciones y recompensas divertidas). Seguido, en la fase de **decoración**, la cual es definida como una colección de objetos, contexto local, los aspectos de los personajes y, un contexto divertido. Finalmente, en la fase de **condiciones de uso** se describe como el juego puede ser utilizado y por cuánto tiempo por parte del estudiantado o jugadores (Gutiérrez-Hernández et al., 2013).

Los mismos investigadores agregan que el enfoque de la arquitectura será un modelo de capas en una colección organizada de objetos, establecidos en un paquete de clases, donde se espera que los elementos del videojuego estén organizados y que se reflejen los aspectos culturales. Con dichos aspectos se propone la arquitectura presente en la [Figura 7.5](#).

En la arquitectura de la [Figura 7.5](#), se evidencia que incluyen roles implícitos para el personal, donde el instructor educativo es el encargado de definir los objetivos pedagógicos, resultados esperados (destrezas, habilidades, desarrollo de competencias o habilidades), el dominio del juego (donde se aplica y usa el juego) y características del usuario. Luego, el artista diseñador de videojuegos se encarga de definir las interacciones y establecer la simulación y progresión del videojuego, así como las interfaces de usuario desde el inicio hasta el fin del videojuego. Seguido, el ingeniero de sistemas se encarga de los aspectos físicos y elementos de comunicación del sistema para que la vista de la interfaz se visualice de forma adecuada en el dispositivo. Por último, se tiene al programador, el cual se encarga desarrollar los módulos e interfaces del videojuego (Gutiérrez-Hernández et al., 2013).



**Figura 7.5 Arquitectura propuesta para juegos serios multiculturales**

**Fuente:** Elaboración propia a partir del texto de (Gutiérrez-Hernández et al., 2013, p.3)

A continuación, se describe brevemente cada nivel presente en la arquitectura de la Figura 7.5:

- ✓ **Capa de contexto cultural** – En este nivel se encuentra la información del usuario y recursos del juego, es donde se van a almacenar la totalidad de datos de usuario y archivos fuente multimedia para aplicarlos en el videojuego.
- ✓ **Aplicación** – Este nivel tiene la función de especificar los elementos genéricos del videojuego con base en la lógica y los recursos disponibles, con el fin de responder y proporcionar una interfaz para la administración de eventos y retroalimentación al usuario.
- ✓ **Decoración** – En este nivel se actualizan los estados de los elementos, mediante la observación de la interacción del usuario utilizando patrones de diseño, centrándose en los aspectos culturales visibles, tales como los colores y

códigos de caracteres. Además, se adaptan las características observables de la cultura para los objetos del videojuego.

- ✓ **Objetos de Juego** – En este nivel se integran los módulos y objetos responsables de ofrecer características propias del juego, al gestionar los recursos lógicos y las vistas entre los componentes y paquetes del videojuego para un escenario.
- ✓ **Escenario** - En este nivel se da la integración de clases relativas al correcto funcionamiento interno del videojuego, tales como los objetos, multimedia, reglas de interacción, así como las vistas consideradas en la clase general del objeto. Por otro lado, se encarga de proporcionar a la simulación en tiempo real la variedad de personajes y elementos de objetos del videojuego.
- ✓ **Interfaz de Usuario** – Este nivel permite mostrar el conjunto de escenarios y objetos del videojuego. Además, se da el procesamiento de datos de entrada del usuario para realizar diferentes interacciones en la historia o niveles de juego del contexto cultural del usuario/jugador.

En la [Tabla 7.7](#), se logra observar la asignación de roles del personal, por nivel y cómo participan en la arquitectura para la gestión del juego serio.

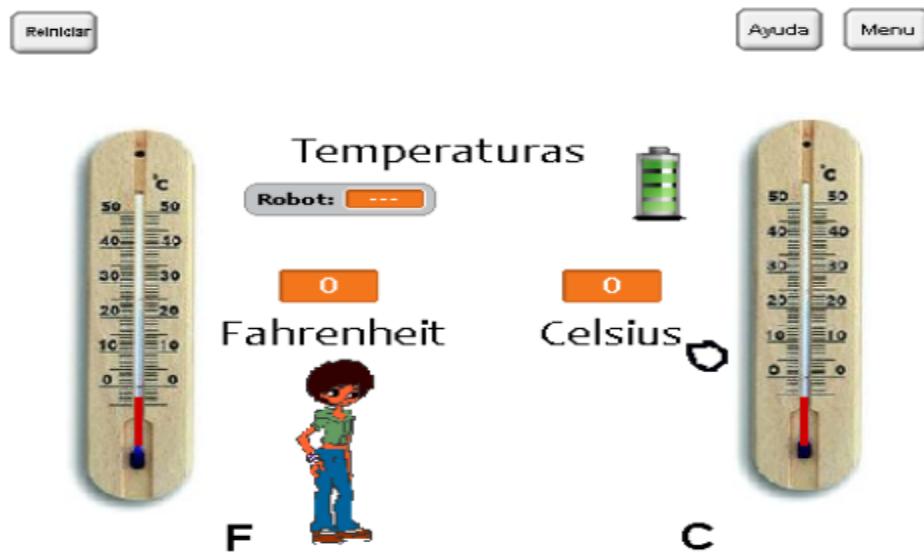
*Tabla 7.7 Relación del personal involucrado en la arquitectura*

Vistas	Personal Involucrado	Nivel Arquitectura
Funcionalidad	Arquitecto, Instructor	1,2,4,6
Rendimiento y Escalabilidad	Artista, Diseñador	2,3,4
Gestión del software	Programador	2,5,6
Topología del Sistema	Ingeniero de Sistemas	6

**Fuente:** Tabla adaptada a partir del texto de (Gutiérrez-Hernández et al., 2013, p.4)

El juego serio multicultural (JSM) desarrollado, es específico para la educación en las ciencias exactas en el área de química. Dicho juego trata el tema de los grados de temperatura en Fahrenheit ( $F^{\circ}$ ) y Celsius ( $C^{\circ}$ ) y puede ser jugado por medio de un navegador web. Al implementar aspectos culturales, se tiene que la temperatura varía dependiendo de la región o el país, por lo que este sería el típico caso de contenidos con aplicación multicultural. Otras modalidades que se le puede

brindar al juego consisten en medir la temperatura del clima para ciertas temporadas de los alimentos, para conocer si pueden ser ingeridos. Igualmente, puede medir la temperatura del cuerpo para determinar alguna enfermedad, entre otros. La Figura 7.6 muestra el prototipo de la interfaz del juego, que abarca todos los niveles de la arquitectura.



**Figura 7.6** Implementación de la arquitectura en el videojuego de temperaturas  
**Fuente:** Figura tomada del texto de (Gutiérrez-Hernández et al., 2013, p.4)

En la Tabla 7.8 se puede observar la representación de los objetos del videojuego establecidos en cada nivel de la arquitectura anterior para la interfaz del videojuego.

**Tabla 7.8** Representación de los niveles de la arquitectura para el caso de estudio

Nivel	Elementos
Interfaz de Usuario	Escenario Juego
Escenario	EstadoEscenarioMenu. EstadoEscenarioJuego. EstadoEscenarioAyuda.
Objetos del Juego	ObjetoTemperatura. ObjetoBarraEnergía. ObjetoFormula. ObjetoPersonaje. ObjetoMensaje. ObjetoAyuda.
Aplicación	ObservadorObjetos. ObservadorEscenario.
Decoración	EstadoJuego. DecoradorObjeto.
Contexto Cultural	ImagenesObjetos.

**Fuente:** Tabla adaptada a partir del texto de (Gutiérrez-Hernández et al., 2013, p.5 )

Al implementar la arquitectura en el caso de estudio, se evidenció que la organización, y principalmente, el desarrollo de videojuegos, se facilita, debido a que dicha arquitectura proporciona una visión amplia de los elementos de los objetos que componen al juego, así como la comunican entre sí (elementos) en los diferentes niveles presentes en la arquitectura. Con ello, se comprende de manera más adecuada qué aspectos culturales se ven reflejados en la arquitectura y cómo son implementados en el juego.

**Tabla 7.9 Resumen - Aspectos de análisis de (Gutiérrez-Hernández et al., 2013)**

Categoría	Criterios	Descripción
Aspectos generales	País de la propuesta arquitectónica. Idioma de la propuesta. Nivel educativo.	México Español. Educación primaria.
Aspectos arquitectónicos	Fundamentación. Tipo de dispositivos. Reutilización de componentes	ArJuS. Aplicación web. No.
Aspectos pedagógicos	Destinatarios de la arquitectura.  Propósito de la arquitectura. Nivel de complejidad.	Desarrolladores, programadores y docentes.  Desarrollo juego serio multicultural. Nivel medio.
Aspectos de análisis	Tipo de producto a desarrollar (juego)  Resultados de utilización obtenidos. Guías o ayudas disponibles.	Juego serio multicultural. No. No.

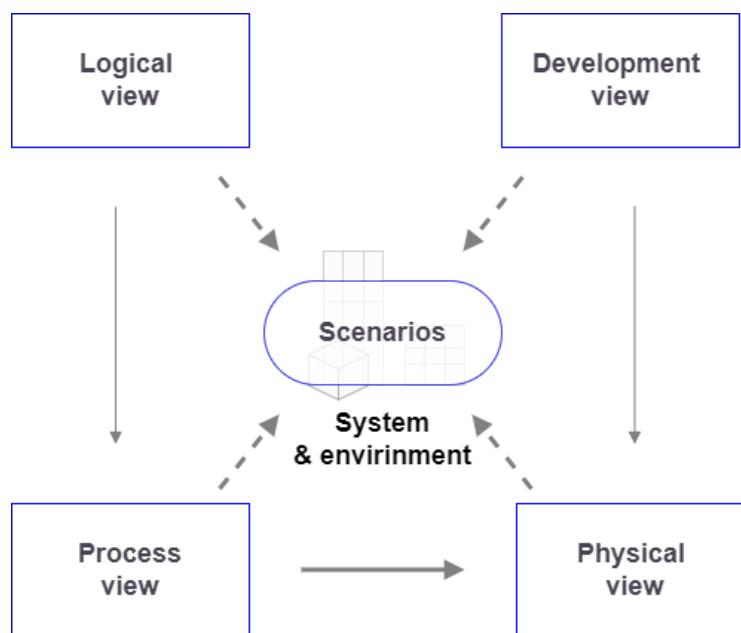
#### 4.16.4. Propuesta de (Söbke & Streicher, 2016)

En **Alemania**, los investigadores (Söbke & Streicher, 2016) realizaron un compendio de varias arquitecturas de software, así como ejemplos para motores de juego. Los investigadores recalcan que la inclusión de personas de diferentes áreas del conocimiento juega un papel importante en la comunicación y la formación de equipos de trabajo en el desarrollo de videojuegos.

El desarrollo de un juego serio a diferencia de uno común es la tarea adicional de implementar el elemento “serio” dentro del juego. Es por ello, que el proceso se puede volver un poco tedioso, debido a que elimina grados de libertad del proceso de diseño, para alcanzar la correcta integración de contenido y juego. Entre todas las diferencias del desarrollo de juegos convencionales, se tiene que los requisitos de hardware para los juegos serios deben ser menores, ya que, usualmente se desean implementar en centros educativos, a pesar de que en algunas ocasiones se

carence de la disponibilidad del hardware más reciente. Las pruebas a los juegos serios no solo comprometen las pruebas de jugabilidad, sino que los pasos de validación se deben incluir para testear que el propósito o intención pedagógica actual del juego se pueda cumplir. Es por ello, que la funcionalidad de monitoreo se debería integrar en el desarrollo de los mismos (Söbke & Streicher, 2016).

Se toma en cuenta la propuesta de la vista 4+1 del modelo de arquitectura propuesto por (Kruchten, 1995; Meng et al., 2010) en donde se proponen cuatro vistas de una arquitectura de software para la descripción de dicho modelo. La primera, relacionada con una vista lógica que se encarga de ilustrar el modelo del objeto. La segunda, vinculada con una vista de proceso que se ocupa de la concurrencia y los problemas de sincronización. La tercera, se relaciona con una vista física que describe el mapeo del software en el hardware y refleja sus aspectos distribuidos. Finalmente, la cuarta vista vinculada con el desarrollo que representa la organización estática del software en su entorno de desarrollo. Dicha vista del modelo de arquitectura se puede observar en detalle la Figura 7.7.

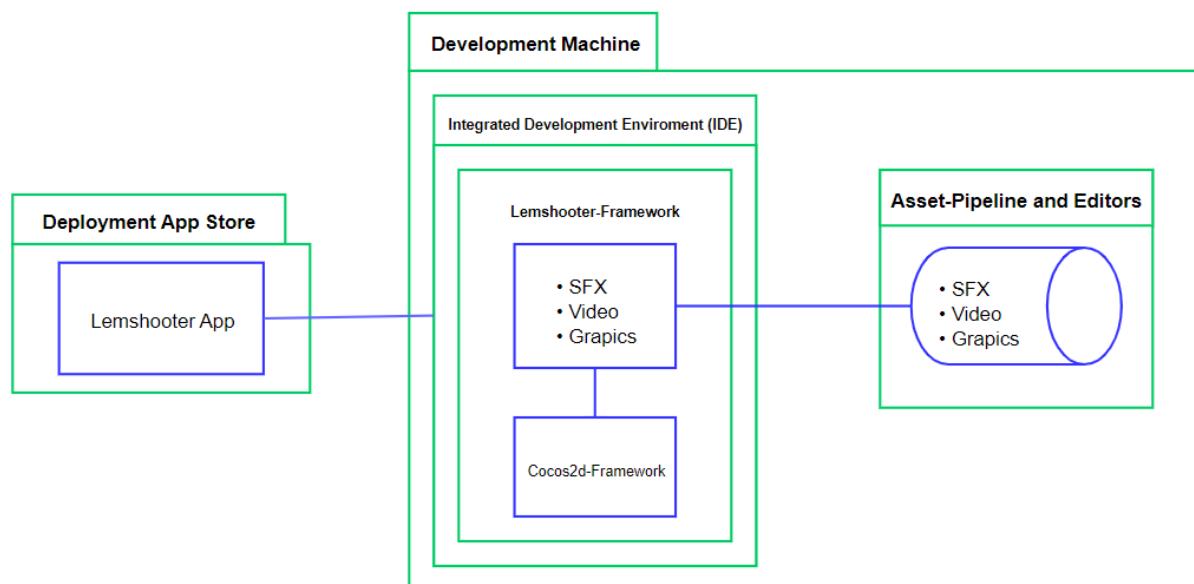


**Figura 7.7** Modelo de la vista 4+1 de una arquitectura por Kruchten  
**Fuente:** Elaboración propia a partir del texto de (Söbke & Streicher, 2016, p.5)

A parte de los motores de juego, existen diversos componentes que son incluidos en las arquitecturas de juegos serios, los cuales podrían ser empleados en

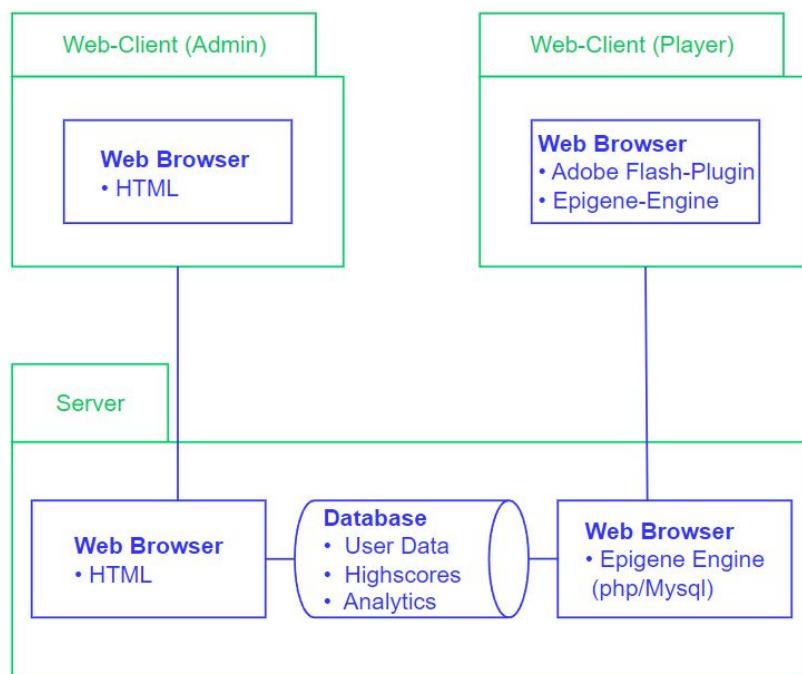
dispositivos móviles, PC, consolas, aplicaciones web, entre otros (Söbke & Streicher, 2016).

- ✓ **Computadoras (PC)** - Un juego serio educativo para PC llamado *Mobility* fue desarrollado con el apoyo del consorcio industrial y gubernamental alemán. Dicho juego consiste en construir ciudades con el enfoque de simulación del tráfico. Cabe destacar que posee una arquitectura simple, debido a que es un juego instalado en una computadora con Microsoft Windows. Su desarrollo no involucró ningún tipo de motor de juego, sino que fue creado con el lenguaje de programación C++ (Söbke & Streicher, 2016).
- ✓ **Dispositivos móviles** - La aplicación *JuraShooter StGB* es una aplicación educativa para estudiantes de derecho. Brinda preguntas de respuesta múltiple para aprender y motivar al estudiantado durante el proceso de respuesta. Fue desarrollada utilizando Cocos2d, un *framework* de desarrollo de *software open source*. Su estructura de software está basada en un framework llamado *LernShooter*, donde se brinda otro set de gráficos, video, efectos de sonido y contenido en la forma de MRQs (Söbke & Streicher, 2016). A continuación, la [Figura 7.8](#) permite mostrar la vista de desarrollo de una aplicación creada por medio de *LernShooter*.



**Figura 7.8** Vista de desarrollo de la arquitectura *LernShooter*  
**Fuente:** Elaboración propia a partir del texto de (Söbke & Streicher, 2016, p.7)

✓ **Aplicaciones web** - Energetika es un ejemplo de los juegos serios basados en la web. Ha ganado el premio más importante de Alemania con respecto a juegos digitales, para la categoría de juegos serios en el año 2011. Este juego ayuda a ilustrar características y consecuencias de diferentes fuentes de energía. Simula temas como el suministro de energía a naciones industriales, problemáticas crecientes como la emisión de CO<sub>2</sub> y repositorios de desechos radioactivos. Su base técnica es el Frameworks basado en Adobe Flash llamado Epigene (Söbke & Streicher, 2016). La vista física de Energetika, se puede observar en detalle en la Figura 7.9.

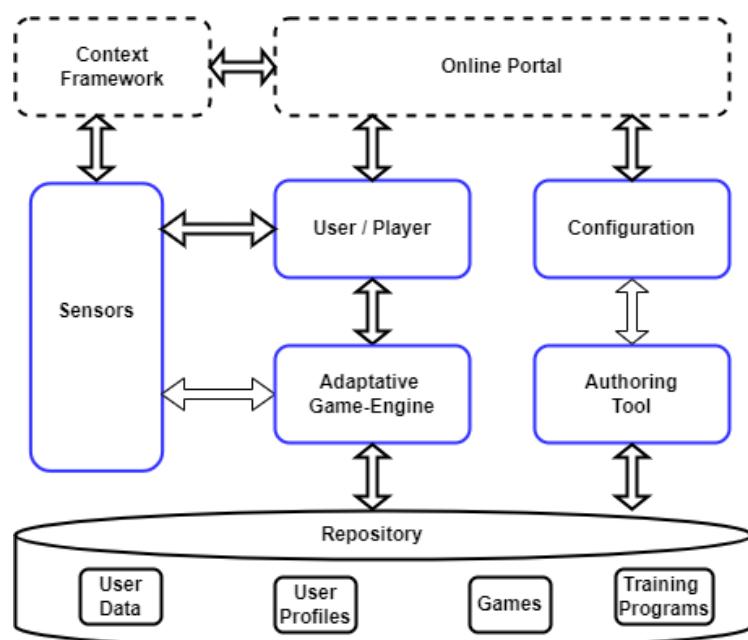


**Figura 7.9** Vista física del juego para aplicación web Energetika

Fuente: Elaboración propia a partir del texto de (Söbke & Streicher, 2016, p.9)

✓ **Consolas** – Según (Söbke & Streicher, 2016) resulta extraño encontrar un juego serio desarrollado para consolas de videojuegos, no obstante, utilizan dispositivos extra de entradas sensoriales, como el Nintendo Wii Remote que facilita juegos deportivos de simulación mediante el uso de un acelerómetro y sensores ópticos. También se encuentra la Microsoft Kinect que rastrea el movimiento de los jugadores mediante una cámara de rango (Söbke & Streicher,

2016). Estos sensores pueden emplearse como componentes de arquitecturas. Un ejemplo de un juego serio que utiliza sensores para procesar información es *Aiming Game* que utiliza biorretroalimentación proporcionada por sensores mediante electroencefalografía y electromiografía para entrenar la regulación de las emociones. El juego *Letterbird* es un ejemplo de un juego “exergame” (juego de ejercicio) que intenta controlar la carga de entrenamiento de resistencia, ya que procesa mediante sensores los datos del atleta (ritmo cardiaco) y el dispositivo de entrenamiento (ritmo de pedaleo, ajustes de resistencia y ergómetro ) (Söbke & Streicher, 2016). En esta línea, los investigadores (Hardy et al., 2015) desarrollaron un Frameworks para juegos serios adaptables para la salud (ver Figura 7.10), el cual es un claro ejemplo de componentes que forman una arquitectura de sistema.



**Figura 7.10** Framework para juegos serios adaptables para la salud  
**Fuente:** Elaboración propia a partir del texto de (Söbke & Streicher, 2016, p.10)

En cuanto a la arquitectura, amplía la vista física con sensores basados en hardware y probablemente más componentes del motor del juego. La vista de desarrollo se amplía con controladores e interfaces de programación (Söbke & Streicher, 2016).

Comúnmente las arquitecturas *web-based* incluyen servidores, los cuales hospedan al juego por sí mismo y lo entregan al navegador web. Además, albergan

herramientas de creación o sistemas de gestión de contenido, así como las bases de datos con contenido, datos de usuario, datos de entrada sensoriales, entre otros (Söbke & Streicher, 2016). En esta línea, otro componente utilizado en el desarrollo de juegos profesionales basados en la web corresponde a una gestión de solicitudes de gestión de cliente-servidor-middleware (*SmartFoxServer* es un ejemplo de un middleware de este tipo). Además, para seleccionar un cliente / servidor apropiado, el modelo es crucial en el caso de mundos virtuales muy frecuentados (Söbke & Streicher, 2016).

En las arquitecturas distribuidas con datos interoperables se requiere de modelos para realizar juegos más dinámicos y modulares. Dichas arquitecturas son ampliamente utilizadas en áreas de aplicación donde los componentes de software no solo se ubican en una computadora, sino en sistemas de redes de computadoras.

En el campo más amplio de entornos de realidad virtual distribuida y simulaciones distribuidas se han propuesto varias arquitecturas, tales como MASSIVE (Greenhalgh & Benford, 1995), DIVE (Frécon & Stenius, 1998) y la Arquitectura de Alto Nivel (HLA, *High Level Architecture*) o su predecesora Simulación Interactiva Distribuida (DIS, *Distributed Interactive Simulation*). También se hace presente la Arquitectura Orientada a vicios (SOA, *Service-Oriented Architecture*) propuesta por (Carvalho, Bellotti, Berta, De Gloria, Gazzarata, et al., 2015) donde los componentes del juego son distribuidos a varios servidores ofreciendo sus servicios por medio de interfaces web.

CIGA es una arquitectura de software propuesta por (van Oijen et al., 2012) donde se conectan sistemas de multi-agente a motores de juego mediante el uso de ontologías como un contrato de diseño. Los investigadores (Jepp et al., 2010) describen una arquitectura basada en agentes para juegos serios modulares. El Frameworks se esfuerza por proporcionar juegos serios con agentes emocionales creíbles para ayudar a los jugadores aprender habilidades y evaluar su desempeño. Es parte de la plataforma TARGET ayudar a los alumnos a entrenar competencias en escenarios de juego. El marco está interconectado con un motor de juego, un sistema de diálogo y un motor narrativo a través de una llamada traducción sincronización y comunicación del manejo del motor.

Tabla 7.10 Resumen - Aspectos de análisis de (Söbke & Streicher, 2016)

Categoría	Criterios	Descripción
Aspectos generales	País de la propuesta arquitectónica. Idioma de la propuesta. Nivel educativo.	Alemania Inglés Educación Superior
Aspectos arquitectónicos	Fundamentación.  Tipo de dispositivos. Reutilización de componentes	SOA, CIGA TARGET, MASSIVE, DIVE, HLA, DIS Móvil, escritorio y web Si
Aspectos pedagógicos	Destinatarios de la arquitectura.  Propósito de la arquitectura. Nivel de complejidad.	Estudiantes, Profesores, Desarrolladores Análisis de otras arquitecturas. Media.
Aspectos de análisis	Tipo de producto a desarrollar (juego)  Resultados de utilización obtenidos. Guías o ayudas disponibles.	Juegos serios: Mobility, JuraShooter StGB, Energetika, Aiming Game, Letterbird. No. No.

#### 4.16.5. Propuesta de (Stavrev et al., 2018)

En **República Checa**, los investigadores (Stavrev et al., 2018) proponen una arquitectura para el diseño de juegos serios, la cual coincide con la tendencia en el campo de los videojuegos. Dicha arquitectura toma en cuenta las restricciones de hardware mencionadas por (Söbke & Streicher, 2016). El objetivo de esta nueva propuesta consiste en hacer que la arquitectura planteada, sea independiente de la entrada de hardware, con el fin de que sea compatible con mayor cantidad de dispositivos.

Los investigadores (Stavrev et al., 2018) retoman la clasificación de juegos serios realizada por (Söbke & Streicher, 2016) la cual considera aspectos relacionados con la perspectiva sobre el diseño, propósito e implementación, a saber:

1. El uso de componentes típicos para la construcción de Juegos serios.
2. Arquitecturas distribuidas.
3. Modelos de datos e interoperabilidad.

El **primer punto de vista** se basa más que todo en la plataforma en que se va a utilizar el juego serio, ya sea dispositivos móviles, juegos en internet, computadora

o consola. Los autores incluyen sensores físicos como la Kinect (Xbox) y el Oculus Rift (Realidad virtual). El **segundo punto** es el de los sistemas descentralizados donde el juego se distribuye en diferentes dominios. Otro aspecto de este punto es el de la naturaleza modular. El **tercer punto de vista** se concentra en el flujo de datos y el uso de estructuras de datos. La idea principal es que los formatos de datos comunes, estructuras de datos y protocolos deben ser utilizados para la creación de la arquitectura de los JS y su contenido (Stavrev et al., 2018).

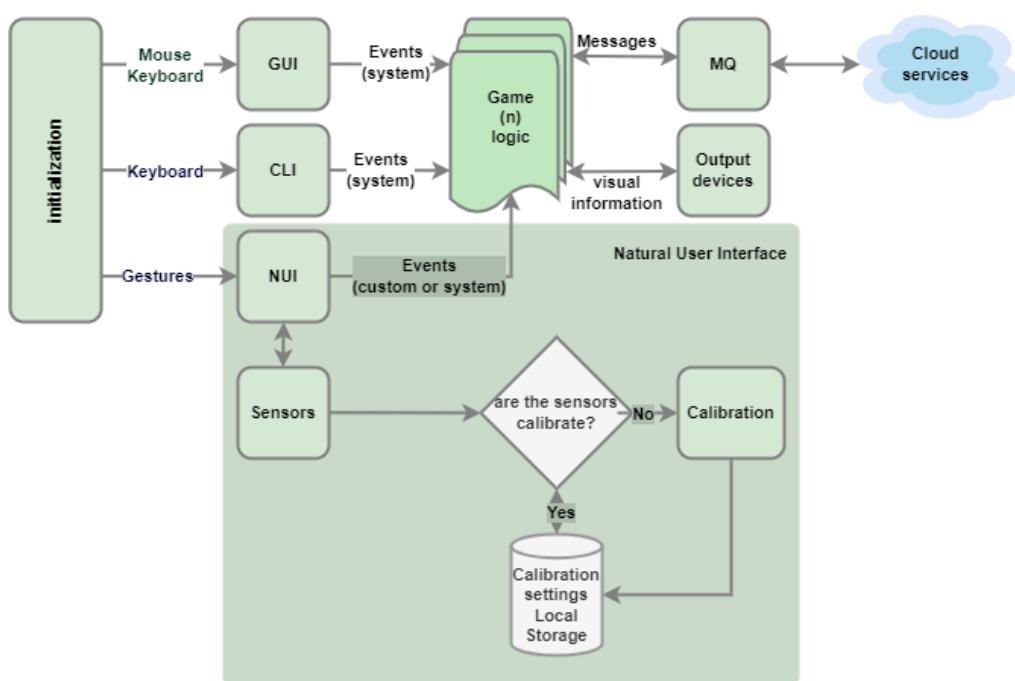
En esta línea, el proyecto RAGE (Van Der Vegt, Nyamsuren, et al., 2016; Van Der Vegt, Westera, et al., 2016) se centra en la reutilización de componentes y activos de juego (*game-assets*), donde se utilizan componentes de juego interoperables (*assets*) que pueden compartirse entre diferentes juegos. Un componente es un objeto del juego que posee cierta funcionalidad, por ejemplo, un activo de vehículo que tiene su conducción y su lógica de dirección. La idea principal de RAGE consiste en que, los componentes puedan ser desarmados y reutilizados por diferentes motores de juego, lenguajes de programación y arquitecturas. Los activos pueden ser comprados o adquiridos en un repositorio central, como la *Unity asset store* o el *Unreal Marketplace*. Desde el punto de vista del software-arquitectónico este proyecto adopta el enfoque de desarrollo basado en componentes (Stavrev et al., 2018).

Los investigadores (Stavrev et al., 2018) definen tres puntos clave con respecto a los principios de una arquitectura para un juego serio:

- ✓ **Diseño modular** - El diseño modular distribuido juega un papel importante en la escalabilidad y la gestión de contenido.
- ✓ **Servicios** - Los servicios y SOA ayudan a integrar diversa información en tiempo real en los juegos.
- ✓ **Sensores** - Los sensores y otros dispositivos hapticos son importantes ya que brindan una interacción única y natural especialmente con fines educativos.

A partir de las consideraciones anteriores, (Stavrev et al., 2018) proponen la Arquitectura Distribuida para Juegos Serios Independiente de Entradas (DiAS: *Distributed Input-Independent Architecture for Serious games*).

La Figura 7.11 muestra la arquitectura propuesta (DiAS) la cual contiene diferentes fases. En la fase de inicio se configuran diferentes componentes, tales como la ausencia o presencia de sensores, la cantidad y tipo de dispositivos de salida, monitores proyectores, entre otros. La fase de inicio es importante en particular, ya que, decidirá cómo el sistema responderá a las interacciones del usuario.



**Figura 7.11** Vista general de DiAS

Fuente: Elaboración propia a partir del texto de (Stavrev et al., 2018, p. 1168)

Tal como se puede observar en la Figura 7.11, se diseñaron tres interfaces genéricas con la función de comunicarse con el sistema:

- ✓ **Interfaz gráfica del usuario (GUI)** - Elementos gráficos tales como botones, menús, textos, entre otros. En principio, esta interfaz está intencionada para ser utilizada por educadores y ocasionalmente por los jugadores.
- ✓ **Interfaz de comando de línea (CLI)** - Interfaz especial intencionada para ser utilizada por solamente un administrador del sistema o un asegurador de

calidad (QA tester). Permite al usuario ingresar comandos predefinidos (activan diferentes funciones o grupos de funciones en la lógica del juego) por medio de la línea de comandos “in-game”

✓ **Interfaz del usuario natural (NUI)** - Interactúa con la lógica del juego

mediante la captura de gestos hechos por los usuarios. Dichos gestos pueden ser aplaudir, menear la cabeza, brincar o caminar, saludar, entre otros. Una vez que son capturados por el sensor, cada gesto será procesado como un evento personalizado o un evento de sistema genérico.

Es importante resaltar que la interfaz GUI es la única que se encenderá por default. Si el inicializador detecta la presencia de sensores conectados al sistema, la NUI también se encenderá. La CLI es una interfaz especial que solo se podrá acceder a ella por medio de un archivo de configuración.

El **módulo de lógica del juego** es el juego real. Cada juego recibe información (en forma de eventos) a través de una de las interfaces de usuario (GUI, CLI, NUI). Hay dos enfoques para disparar eventos de entrada: disparando un evento personalizado o disparando un evento de SO (Sistema operativo) emulado. Cada uno de estos métodos tiene sus pros y sus contras. La ventaja de traducir un gesto a un evento de sistema / SO es que la arquitectura se vuelve más genérica. De esa manera, la lógica del juego se puede desacoplar de la interfaz de usuario.

Los eventos mismos pueden ser eventos COM en sistemas Windows o mensajes D-bus en sistemas Unix. Si la lógica del juego es programada para escuchar eventos del sistema, recibirá eventos del sistema. Por ejemplo, el juego escucha para un evento *mouseClick*. El usuario aplaude con las manos generando así un gesto, la NUI captura el gesto (a través de un sensor), lo consume y dispara un evento *mouseClick*. El proceso de reconocimiento, procesamiento y mapeo de gestos a un evento del sistema consume más tiempo.

El otro alcance de usar la NUI es disparar un evento personalizado para cada gesto. La velocidad de disparo y procesamiento de un evento personalizado será mayor, pero la lógica del juego necesita saber de antemano qué tipo de evento personalizado debe escuchar, agregando así una dependencia a la NUI.

Para que la arquitectura propuesta sea adaptable, escalable e independiente, fue diseñada para que se pueda utilizar con diferentes configuraciones, incluidos

diferentes sensores de entrada. Se puede utilizar el diseño actual del sistema con (pero no limitado a) *Kinect*, cámaras de profundidad genéricas, *Orbbec Astra*, *Leap motion* e *Intel RealSense*. La arquitectura propuesta permite el desacoplamiento de la interfaz de usuario (que puede ser sensorial) de la lógica del juego real (jugabilidad).

Cualquier juego que se pueda utilizar con un mouse y un teclado se puede jugar con una entrada sensorial. La calibración del sensor suele ser un proceso tedioso y que requiere tiempo. En la arquitectura propuesta por (Stavrev et al., 2018), se realizan esfuerzos para minimizar el tiempo de calibración, para ello, se utiliza el "intercambio de configuraciones de calibración", donde la configuración de calibración del sensor se puede compartir entre diferentes juegos e instancias del mismo.

La inicialización del módulo de la arquitectura DiAS se puede implementar y construir en todos los juegos que admitirán una NUI. De esa manera, los sensores, conectados al sistema, se pueden calibrar una vez (en un juego); sus ajustes (dependiendo del sensor, rango desde el usuario, tipos de interacciones, resolución, entre otros) se pueden escribir en un almacenamiento local y ser reutilizados para otro juego, sin la necesidad de recurrir al proceso de calibración nuevamente.

DiAS es una arquitectura con un enfoque abstracto arquitectónico para juegos serios que combina principios de diseño, donde se propone un enfoque principalmente en tres puntos: interfaces independientes de dispositivos de entrada, adaptabilidad y el uso de servicios para funcionalidades dentro del juego.

La independencia del dispositivo en DiAS se logra de dos maneras: disparando y procesando eventos o emulando eventos del sistema nativo. La adaptabilidad se logra compartiendo opciones de configuración que pueden usar diferentes instancias de juego. La conexión a servicios de terceros se logra mediante un mecanismo de publicar-suscribirse mediante el uso de un agente de MQ como middleware. Además, los módulos y servicios reales que solicitan los datos de servicios de terceros se alojan en la nube.

En suma, la descentralización y modulación es útil para que el juego serio creado pueda correr en hardware de bajo costo.

**Tabla 7.11 Resumen - Aspectos de análisis de (Stavrev et al., 2018)**

Categoría	Criterios	Descripción
Aspectos generales	País de la propuesta arquitectónica. Idioma de la propuesta. Nivel educativo.	República Checa Español Educación Superior
Aspectos arquitectónicos	Fundamentación. Tipo de dispositivos. Reutilización de componentes	SOA Aplicación web Si
Aspectos pedagógicos	Destinatarios de la arquitectura. Propósito de la arquitectura. Nivel de complejidad.	Profesores, Desarrolladores Formación de competencias Medio
Aspectos de análisis	Tipo de producto a desarrollar (juego) Resultados de utilización obtenidos. Guías o ayudas disponibles.	No se menciona ninguno en concreto No No

#### 4.16.6. Propuesta de (Flores et al., 2019)

En México, el Instituto para la Evaluación de la Educación, aplicó en el 2018 las pruebas (PLANEA) a 100.000 estudiantes de 6to año. Estas pruebas tenían el propósito de evaluar los aprendizajes en diversas áreas. En este sentido, los resultados obtenidos en el área de matemáticas fueron preocupantes, ya que, más de la mitad del estudiantado se encontraba en un nivel insuficiente (El 59% del estudiantado poseía un nivel insuficiente, el 18% un nivel básico, el 15% un nivel satisfactorio y, el 8% un nivel sobresaliente). Por ello, como estrategia didáctica, se diseñaron e implementaron diferentes técnicas de aprendizaje, entre ellas, el uso e integración de los juegos serios en los procesos formativos del estudiantado.

Los investigadores (Flores et al., 2019) se proponen crear un juego serio que incorpore esas técnicas de aprendizaje, de igual forma, proponen la arquitectura para dicho juego. La intención pedagógica del juego consiste en potenciar el rendimiento del estudiantado de sexto año en el área de Matemáticas.

Para llevar adelante el desarrollo del juego, los investigadores (Flores et al., 2019) proponen una clasificación de juegos serios orientados a resultados (Tabla 7.12). Para efectos del juego serio por desarrollar para el aprendizaje de las matemáticas, se consideró enfocarlo en la clasificación de educación.

**Tabla 7.12 Clasificación de los juegos serios orientados a resultados**

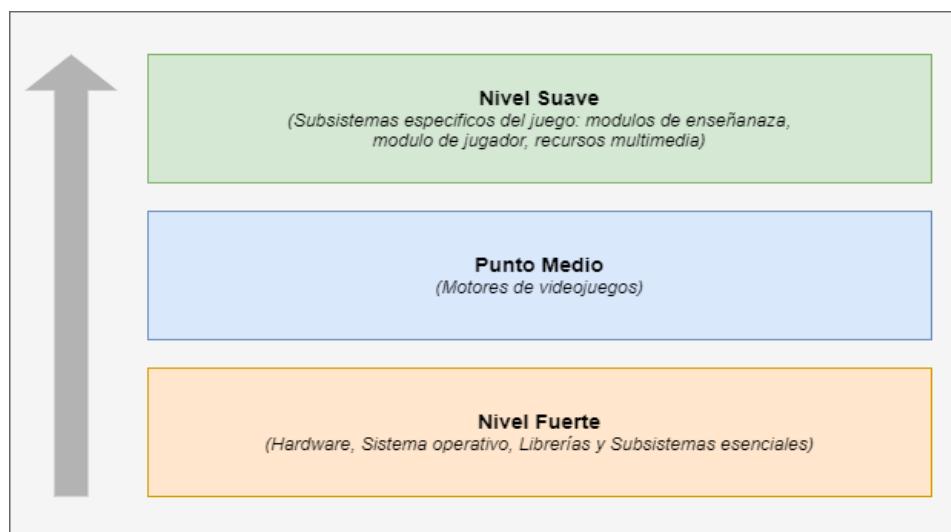
Clasificación	Descripción
Educación	Ejercicios que comunican y evalúan habilidades, conceptos y capacidades.

Clasificación	Descripción
Toma de decisiones	Proporcionan experiencias que construyen marcas que influyen en formar una opinión.
Simulación	Proporcionan insignias puntajes y otras recompensas basadas en el trabajo.

**Fuente:** Tabla tomada y adaptada del texto de (Flores et al., 2019, p. 6)

Los juegos serios usualmente se basan en la simulación de un contenido de aprendizaje que debe ser representado desde el punto de vista del diseño de un juego serio, el diseño puede lograrse por medio de una aproximación global, llamada aproximación endógena, la cual consiste en que el contenido de aprendizaje esté fuertemente ligado con la estructura general del juego y sus reglas. Para el diseño de estos juegos, la aproximación endógena es un buen enfoque, ya que la jugabilidad se encuentra integrada con el contenido de aprendizaje de una manera natural, además de tener una importancia contextual, donde el jugador se encuentra en el centro del proceso de desarrollo, por lo que el juego serio puede contar con objetivos de aprendizaje más complejos (Flores et al., 2019).

Dicho lo anterior, la arquitectura de un juego serio con una aproximación endógena se puede definir en dos niveles o arquitecturas: la del nivel bajo y el alto. El nivel o arquitectura fuerte es aquella que concentra el hardware y un sistema operativo para que los diseñadores, mediante un motor de juegos, creen un juego concreto. En la [Figura 7.12](#), se observa la arquitectura general de un juego serio.



**Figura 7.12 Arquitectura general de un juego serio**

**Fuente:** Elaboración propia a partir del texto de (Flores et al., 2019, p. 7)

El juego serio tiene como objetivo que los estudiantes adquieran o refuerzen sus conocimientos en matemáticas básicas, por ello, es un juego de tipo educativo,

en el que el jugador toma el rol de un navegante que se vio envuelto en una extraña tormenta que lo hizo perder el conocimiento y su nave encalla en una isla. Para salir de esta situación, el navegante debe encontrar las 6 piezas faltantes de su embarcación, para ello, el jugador se enfrentará a diversos retos que exponen desafíos matemáticos específicos que van acorde a las competencias evaluadas en la prueba PLANEA. Los módulos empleados para la arquitectura y la relación entre ellos se pueden observar en la [Figura 7.12](#) y [Figura 7.13](#), respectivamente.



**Figura 7.13 Arquitectura propuesta para un juego serio basado en retos**

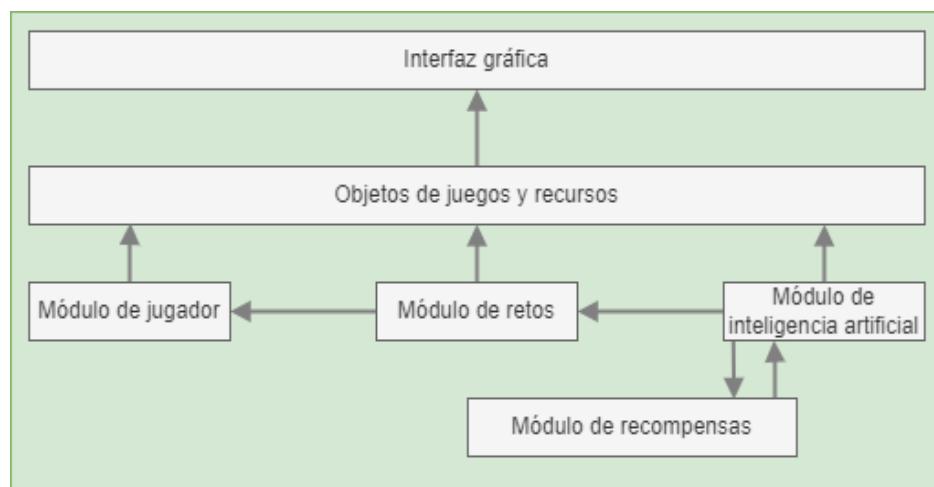
Fuente: Elaboración propia a partir del texto de (Flores et al., 2019, p. 8)

Acorde con (Flores et al., 2019) los módulos de la [Figura 7.13](#) se definen como:

- ✓ **Interfaz gráfica** - Representación visual de los elementos gráficos y auditivos del juego.
- ✓ **Módulo del jugador** - Representa al jugador dentro del juego. También almacena los atributos del jugador (nombre, avance y desempeño durante el juego).
- ✓ **Módulo de retos** - Provee los retos de matemática basados en las pruebas PLANEA al jugador. Dichos retos son misiones en donde el jugador se

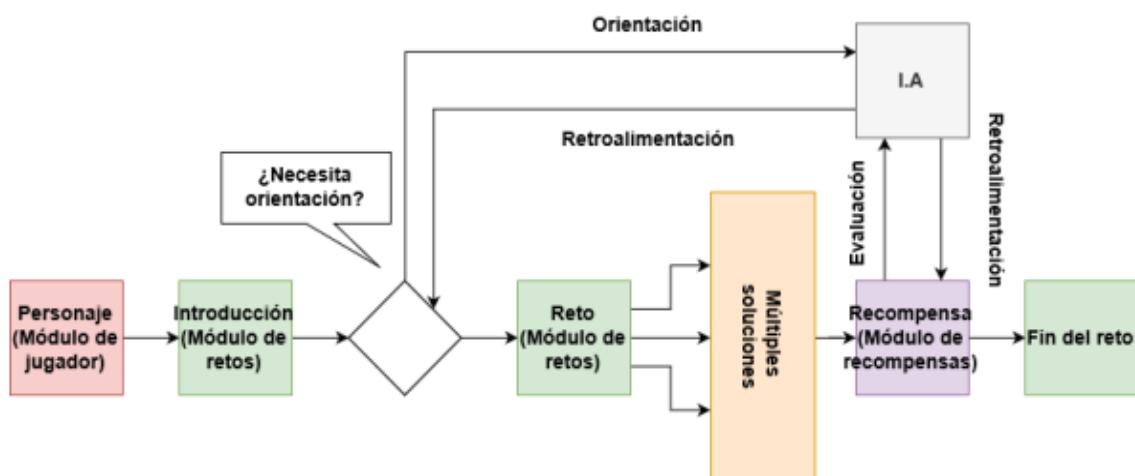
enfrentará a una problemática con múltiples soluciones, en donde el jugador deberá probar su destreza y presentar su solución.

- ✓ **Módulo de recompensas** - Se implementa la evaluación del reto de forma sumativa, además de que se encarga de la evaluación de la solución presentada por el jugador al final de cada reto para proporcionar una recompensa y una realimentación.
- ✓ **Objetos y recursos del juego** – Se almacenan todos los elementos pertenecientes al juego en su forma lógica como por ejemplo el jugador, los escenarios, enemigos y elementos necesarios que conforman los retos.
- ✓ **Módulo de inteligencia artificial** – En él ocurren varias tareas como la de implementar la evaluación formativa del estudiante proporcionando retroalimentación que facilite el aprendizaje durante todo el reto. Además, dependiendo de la evaluación hecha por el módulo de recompensas, se permitirá un aumento o disminución de la dificultad del juego.



**Figura 7.14 Interrelación de los módulos de la arquitectura propuesta**  
**Fuente:** Elaboración propia a partir del texto de (Flores et al., 2019, p. 9)

A continuación de muestra el diagrama de secuencia que seguirá la arquitectura, así como las fases de progresión del juego:



**Figura 7.15 Diagrama de secuencia de la arquitectura propuesta**  
**Fuente:** Figura tomada del texto de (Flores et al., 2019, p. 10)

**Tabla 7.13 Fases de progresión del juego serio basado en la arquitectura propuesta**

Módulo	Fase de jugabilidad	Fase de aprendizaje
Módulo de jugador	El jugador configura el personaje que controlará en el juego	Aprende a configurar el juego
Módulo de retos	Recibe una introducción de la situación específica del nivel del juego y sus objetivos. El jugador intenta resolver el reto propuesto	Aprende las mecánicas del juego y cómo realizar la interacción específica del nivel. El jugador se somete a la experimentación de soluciones dentro del reto.
Módulo de recompensas	Se evalúa el desempeño del jugador y se determina la recompensa apropiada	El jugador aprende de sus errores en caso de fallo para obtener una mejor recompensa.
Módulo de inteligencia artificial	El jugador obtiene información valiosa y consejos de cómo pasar el nivel del juego dependiendo de su desempeño. Además, aumenta o disminuye la dificultad de juego.	El jugador recuerda los consejos del avatar y los aplica al jugar de nuevo el nivel.

**Fuente:** Figura tomada y adaptada del texto de (Flores et al., 2019, p. 10)

La arquitectura propuesta permite la inclusión de módulos de retos, recompensas, de inteligencia artificial y del jugador, de los cuales, al interactuar entre ellos, con los objetos y recursos del juego, permiten al jugador una experiencia única de acuerdo con sus conocimientos.

**Tabla 7.14 Resumen - Aspectos de análisis de (Flores et al., 2019)**

Categoría	Criterios	Descripción
Aspectos generales	País de la propuesta arquitectónica. Idioma de la propuesta.	México Español

**CAPÍTULO VIII. PROPUESTA METODOLÓGICA Y ARQUITECTÓNICA PARA EL DISEÑO DE JS**

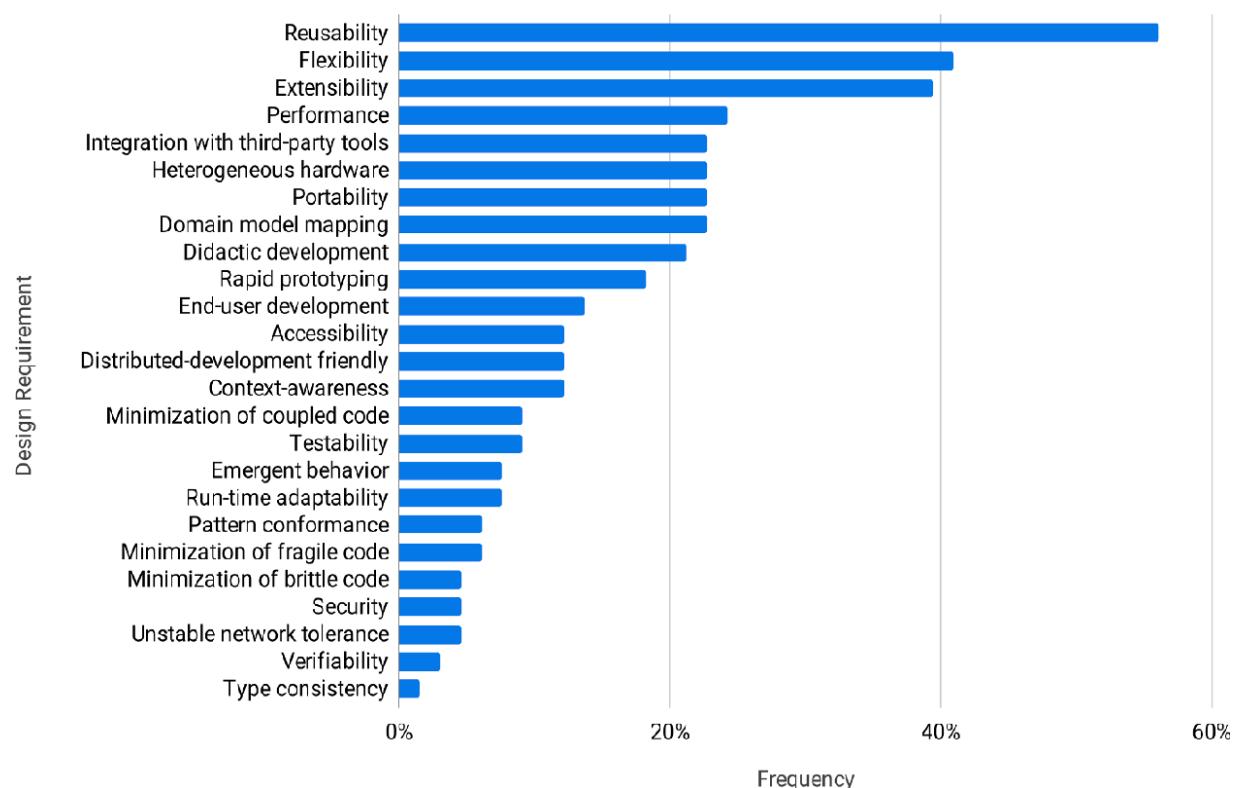
<b>Categoría</b>	<b>Criterios</b>	<b>Descripción</b>
Aspectos arquitectónicos	Nivel educativo.	Educación Primaria
	Fundamentación.	Arquitectura suave, juego serio basado en retos.
	Tipo de dispositivos. Reutilización de componentes	Escritorio No
Aspectos pedagógicos	Destinatarios de la arquitectura. Propósito de la arquitectura.	Profesores, desarrolladores Ser utilizada para crear un juego serio basado en retos de matemáticas para mejorar el desempeño de los estudiantes
	Nivel de complejidad.	Medio
Aspectos de análisis	Tipo de producto a desarrollar (juego)	Juego serio basado en retos de matemáticas para mejorar el desempeño de los estudiantes
	Resultados de utilización obtenidos. Guías o ayudas disponibles.	No No

#### 4.16.7. Propuesta de (Mizutani et al., 2021)

En **Brasil**, los Investigadores (Mizutani et al., 2021) analizaron 36 estudios que abarcan arquitecturas relacionadas con videojuegos, para ello, realizaron una documentación de reflexiones y compromisos entre los requisitos de diseño, prácticas y restricciones. Según (Mizutani et al., 2021), la singularidad de la mecánica del juego, los hace únicos, lo cual reduce la posibilidad de reutilización del software, sin embargo, al utilizar una arquitectura de software de alto nivel para la mecánica del juego, facilitaría la reutilización del software.

Para el estudio, se definieron 3 preguntas de investigación acordes a la relación entre la arquitectura del software y la implementación de la mecánica del juego digital, a saber: **¿Qué desafíos de diseño de software enfrentan los investigadores al implementar la mecánica del juego?**

Los hallazgos obtenidos muestran que la investigación que involucra la arquitectura de software de la mecánica del juego favorece la versatilidad del diseño y, en menor medida, el rendimiento, tal como se puede observar en la Figura 7.16.



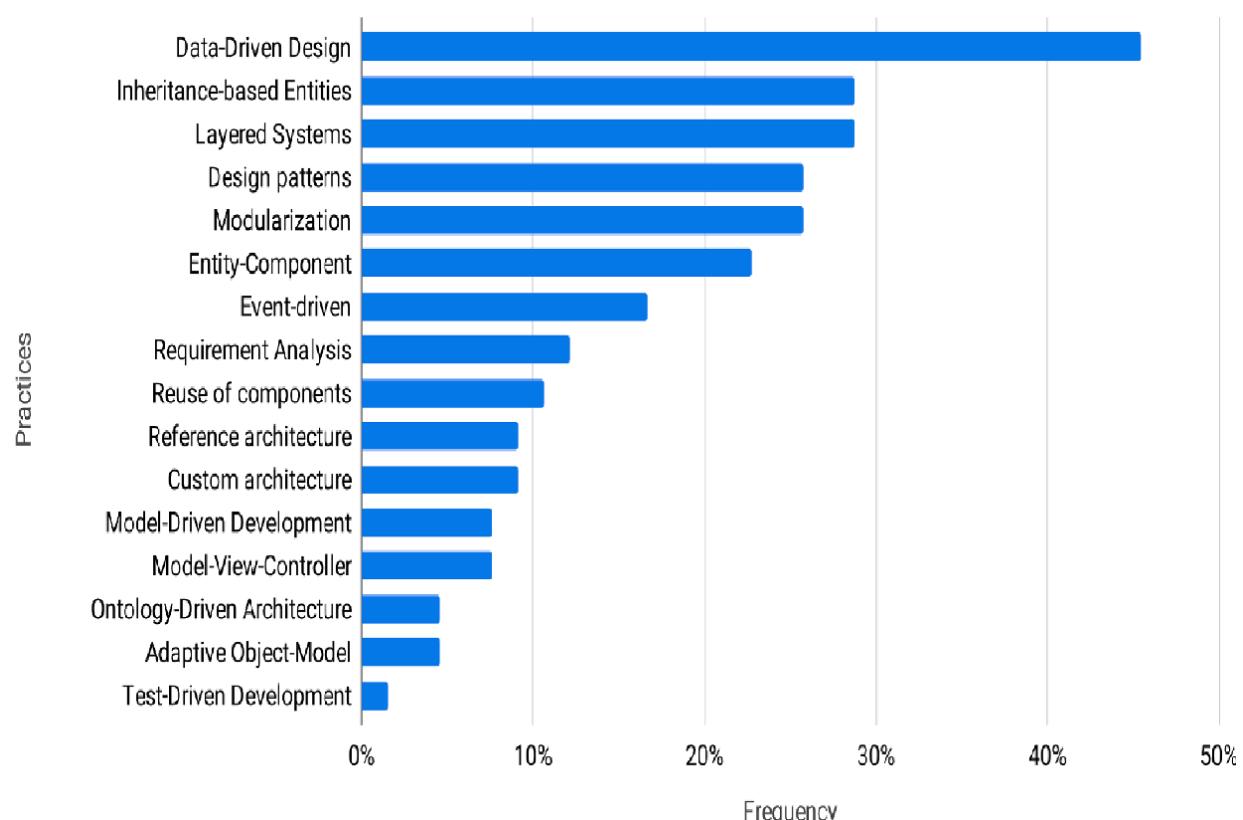
**Figura 7.16 Frecuencia de estudios seleccionados con cada requisito de diseño de SW**

**Fuente:** Figura tomada del texto de ( Mizutani et al., 2021, p.8)

Los resultados sugieren que la compatibilidad con cambios accesibles e inmediatos en la mecánica de juego es una característica valiosa para los investigadores, que generalmente carecen del tamaño de un equipo y la experiencia de las empresas de juegos, por otro lado, se resalta que las soluciones de investigación a menudo conducen como máximo a un prototipo, posiblemente optado por menos estabilidad y robustez en comparación con los juegos listos para producción de la industria.

En relación con la segunda pregunta de investigación, relacionada con: **¿Qué prácticas y patrones de arquitectura de software utilizan los investigadores para la implementación de la mecánica del juego?**

En este caso, los investigadores obtuvieron como resultados que el desarrollo basado en datos, la modularización y las arquitecturas en capas siguen siendo las prácticas de elección en los estudios tanto en la academia como en la industria. Dichos resultados se pueden observar en la [Figura 7.17](#).



**Figura 7.17 Estudios seleccionados con diferentes prácticas arquitectónicas**

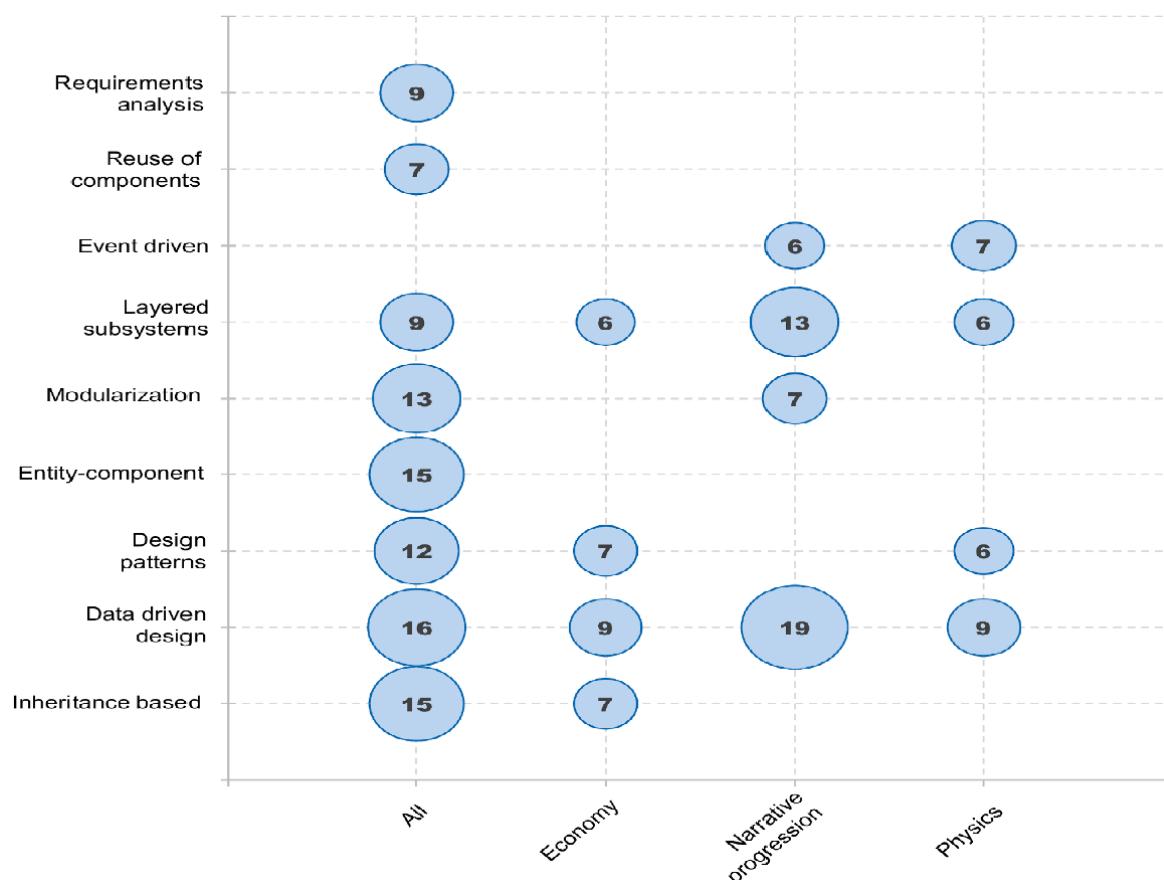
Fuente: Figura tomada del texto de (Mizutani et al., 2021, p.9)

Mizutani et al. (2021) indican que el diseño basado en datos prevaleció principalmente en comparación a las otras prácticas, sin embargo, resaltan que la mayoría de las veces solo se investigó superficialmente, es decir, se aplicó para ajustarse a escenarios específicos, pero no se estudió como una técnica más general, al mismo tiempo, señalan que el patrón entidad-componente-sistema y el uso de patrones de diseño en general, son soluciones confiables.

Con relación a la pregunta de investigación orientada a indagar **¿Qué tipos de mecánica requieren con mayor frecuencia, en la investigación, el uso de prácticas y patrones arquitectónicos, y cuáles son?**

Se obtuvo que la mayoría de los estudios generalizaron la mecánica desarrollada, por lo que solo se pudieron encontrar algunas relaciones entre los tipos de mecánica, requisitos y prácticas de diseño correspondientes, por otro lado, se evidenció la falta de terminología clara para referirse a las partes de la implementación de un juego que se refieren a sus mecánicas.

Finalmente, Mizutani et al. (2021) señalan que la mecánica de la física tiene poca intersección con el campo de la arquitectura de software en general, por ende, requiere el mayor rendimiento. Por otro lado, las mecánicas narrativas y económicas se investigan por separado principalmente como medio para apoyar la educación o la innovación tecnológica. Dichas intersecciones se pueden observar en la [Figura 7.18](#).

**Figura 7.18 Frecuencia entre mecánica del juego y prácticas arquitectónicas**

**Nota:** Intersección entre mecánica del juego (eje horizontal) y prácticas arquitectónicas (eje vertical)

**Fuente:** Figura tomada del texto de (Mizutani et al., 2021, p.12)

**Tabla 7.15 Resumen - Aspectos de análisis de (Mizutani et al., 2021)**

Categoría	Criterios	Descripción
Aspectos generales	País de la propuesta arquitectónica. Idioma de la propuesta. Nivel educativo.	Brasil. Inglés. Otro: Industria.
Aspectos arquitectónicos	Fundamentación. Tipo de dispositivos. Reutilización de componentes	Mecánicas de juego. Móvil, escritorio. Si.
Aspectos pedagógicos	Destinatarios de la arquitectura. Propósito de la arquitectura. Nivel de complejidad.	Desarrolladores. Desarrollo Bajo.
Aspectos de análisis	Tipo de producto a desarrollar (juego) Resultados de utilización obtenidos. Guías o ayudas disponibles.	No. No. No.

#### 4.16.8. Propuesta de (Olsson et al., 2015)

En **Suecia**, los investigadores (Olsson et al., 2015), propusieron el uso del patrón arquitectónico Modelo-Vista-Controlador (MVC), para facilitar la transición a nuevas tecnologías o a diferentes plataformas mediante la separación del código de

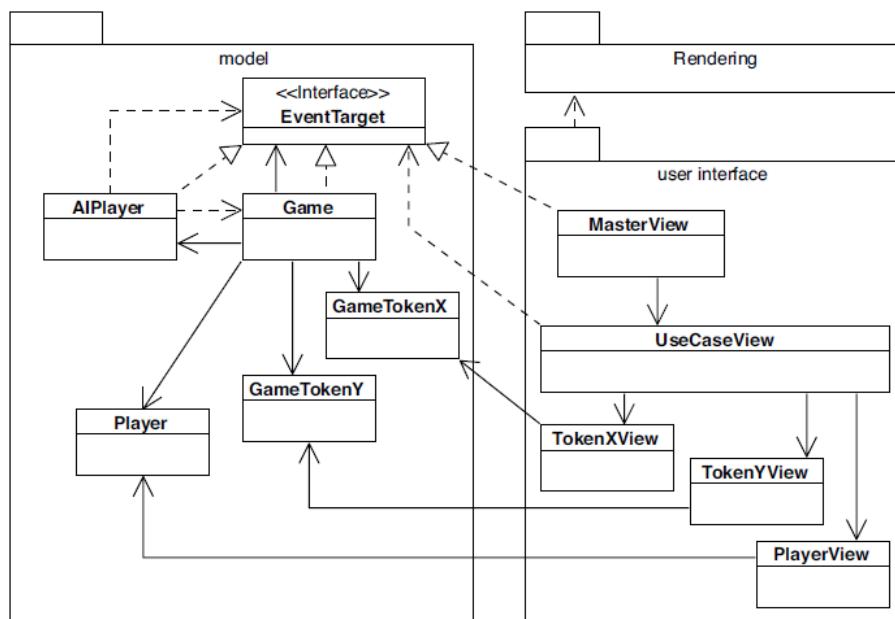
juego del código de renderizado. Definieron un modelo con el objetivo de evaluar la calidad entre las diferentes implementaciones de este patrón arquitectónico en cinco proyectos de juegos desarrollados por el mismo equipo para la plataforma Windows.

Para estudiar las variantes y las diferencias de implementación en el patrón arquitectónico MVC en los diferentes juegos, (Ollsson et al., 2015) realizan un análisis sistemático del código fuente y un gráfico de dependencias, por otro lado, definen un modelo de calidad para estudiar si existen diferencias en la calidad del software. Por último, refactorizan uno de los juegos para utilizar una versión más reciente de la arquitectura, con el objetivo de comparar la calidad del software antes y después de la refactorización.

Los juegos analizados durante la investigación son: Time Breaker, Frontline, T.W.T.P.B, Gears of Love y Hero. A continuación, se describe brevemente los resultados obtenidos de los cinco juegos estudiados (Ollsson et al., 2015):

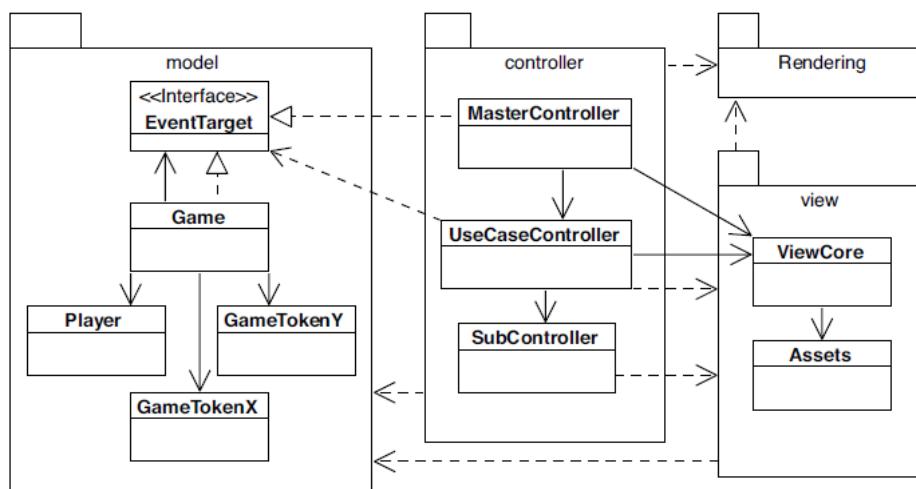
- ✓ **Time Breaker y Frontline** - El diseño es similar en ambos juegos, no se mostraron dependencias directas entre el componente modelo y la interfaz de usuario, además, las interfaces de usuarios consisten en pocas clases grandes que implementan tanto el renderizado como la interacción en los principales escenarios de uso. Cabe resaltar que, el mayor acoplamiento y parte del código duplicado se encuentra en la interfaz de usuario.

Tal como se puede apreciar en la [Figura 7.19](#), ambos juegos tienen una clara división entre los modelos e interfaces de usuario, además, hacen uso del patrón de diseño de software *Observer* (*EventTarget*), encargado de encapsular la comunicación entre los dos componentes principales.



**Figura 7.19** Diagrama de clases sintetizado para *TimeBreaker* y *Frontline*  
**Fuente:** Figura tomada del texto de (Ollsson et al., 2015, p. 10)

✓ **T.W.T.P.B** - Las clases de la interfaz de usuario son las que presentan un mayor acoplamiento. Se evidencia una clara separación entre el modelo y la interfaz de usuario. Se utiliza el mismo patrón de diseño *Observer* de *Time Breaker* y *Frontline* (ver [Figura 7.20](#)).



**Figura 7.20** Diagrama de clases T.W.T.P.B  
**Fuente:** Figura tomada del texto de (Ollsson et al., 2015, p. 11)

✓ **Hero y Gears of Love** - Las clases de interfaz de usuario presentan el mayor acoplamiento en ambos juegos, seguidamente las clases controladoras

tienen el segundo mayor acoplamiento en Gears of love y las clases de modelos en Hero. Existe una estricta separación entre los tres componentes principales (interfaz de usuario, controlador, modelo) (ver Figura 7.21).

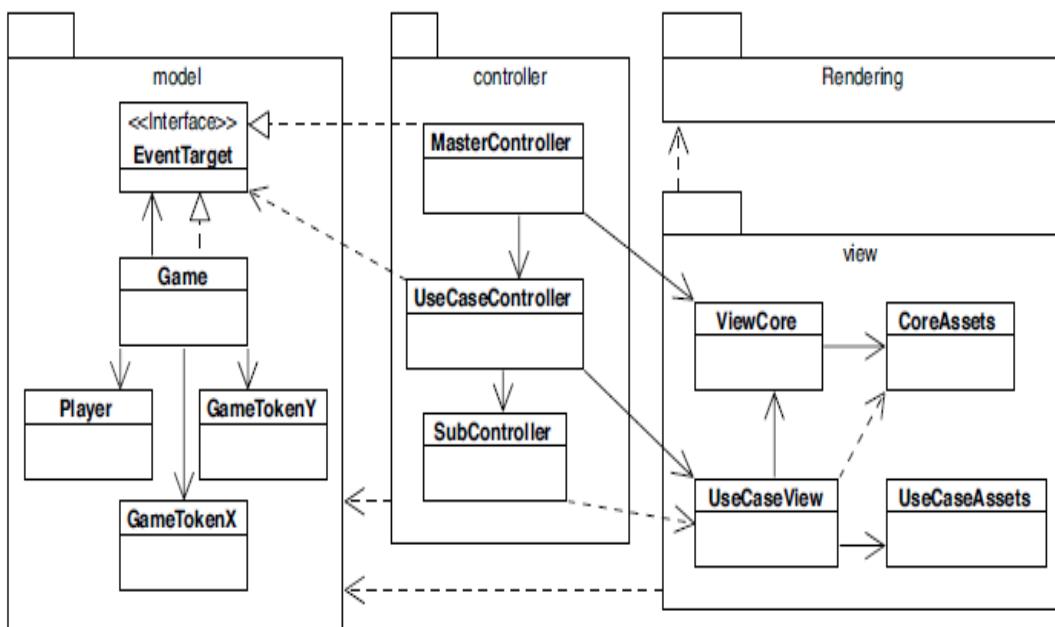


Figura 7.21 Diagrama de clases sintetizado de Hero y Gears of Love

Fuente: Figura tomada del texto de (Ollsson et al., 2015, p. 11)

Los investigadores (Ollsson et al., 2015), realizaron un proceso de refactorización al juego T.W.T.P.B. En dicho proceso, se identificaron 23 tipos en la interfaz de usuario que debían ser refactorizados, los cuales pueden ser observados en la Tabla 7.16.

Tabla 7.16 Refactorizaciones en T.W.T.P.B

Refactorización	Tipos	Acción
Tipo de división	3	Dividir en vista y controlador separados
Tipo de movimiento	6	Mover todo el tipo al componente correcto
Función de desplazamiento	4	Mover la función al tipo correcto
Tipo de encapsulado	10	Mover el tipo al tipo padre

Fuente: Tabla tomada y adaptada a partir del texto de (Ollsson et al., 2015, p. 12)

Las refactorizaciones que presentaron mayores problemas se subdividieron en tipos o categorías que permitieran encapsular de una mejor manera las responsabilidades del controlador y la vista.

La Tabla 7.17 presenta las tres variantes del patrón arquitectónico MVC en las implementaciones.

**Tabla 7.17 Clasificación del proyecto final y la arquitectura de la UI**

Proyecto	Arquitectura de la interfaz de usuario
Hero	Controlador supervisor de grano fino capa de vista estricta
Gears of Love	Controlador supervisor de grano grueso capa de vista estricta
New T.W.T.P.B.	Controlador supervisor de grano fino capa de vista estricta
Frontline	Modelo de separación de la interfaz de usuario
Time Breaker	Modelo de separación de la interfaz de usuario
T.W.T.P.B.	Controlador supervisor, capa de vista relajada

**Fuente:** Tabla adaptada y traducida a partir del texto de (Ollsson et al., 2015, p. 12)

Acorde a los resultados obtenidos y mostrados en la Tabla 7.17, el modelo de calidad muestra que las distintas implementaciones de MVC tienden a dar diferentes calidades de implementación. El controlador supervisor con capa de vista estricta tuvo la mejor calidad de software. Por otra parte, (Ollsson et al., 2015) señalan que del análisis realizado a la utilización del patrón arquitectónico en los proyectos de juegos, se identifican buenas prácticas:

- ✓ Las clases del modelo sólo deberían llevar la funcionalidad y el estado relacionados con las reglas del juego.
- ✓ Los controladores deberían contener el estado de la vista que se necesita para la interacción (por ejemplo, la sincronización entre diferentes estados). Y, si el estado no es necesario para la interacción, se debería considerar el colocar el estado en una vista en su lugar.
- ✓ Utilizar vistas de grano fino que den servicio a un controlador en lugar de una gran vista-interfaz que dé servicio a todos los controladores.
- ✓ Utilizar dos interfaces para separar la comunicación entre el modelo y la interfaz de usuario: una para los eventos del juego y otra para la entrada.
- ✓ Dejar que los controladores y las vistas lean la información en el modelo, pero todas las solicitudes de cambio de datos deberían ir a través de la interfaz de entrada.

**Tabla 7.18 Resumen - Aspectos de análisis de (Olsson et al., 2015)**

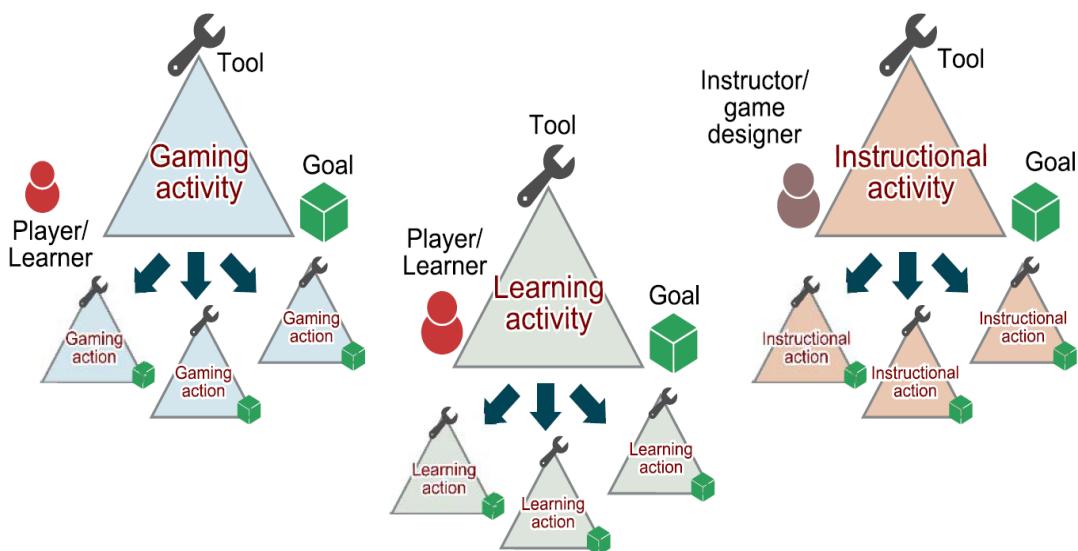
Categoría	Criterios	Descripción
Aspectos generales	País de la propuesta arquitectónica. Idioma de la propuesta. Nivel educativo.	Suecia. Inglés. Otro: Industria.
Aspectos arquitectónicos	Fundamentación. Tipo de dispositivos. Reutilización de componentes	Model View Controller (MVC). Escritorio, Móvil y web. Sí.
Aspectos pedagógicos	Destinatarios de la arquitectura. Propósito de la arquitectura. Nivel de complejidad.	Desarrolladores. Aumentar la calidad del software. Medio.
Aspectos de análisis	Tipo de producto a desarrollar (juego) Resultados de utilización obtenidos. Guías o ayudas disponibles.	Videojuego común. No. No.

#### 4.16.9. Propuesta de (Carvalho, Bellotti, Hu, et al., 2015)

En Italia, los investigadores (Carvalho, Bellotti, Hu, et al., 2015) proponen el uso de la arquitectura orientada a servicios (SOA), como medio para disminuir los costes y el tiempo de desarrollo de los juegos serios (SG). Dicha arquitectura es evaluada por los investigadores mediante un modelo llamado, Modelo de Juegos Serios Basados en la Teoría de la actividad (ATMSG), para identificar los componentes relevantes existentes que pueden ser reutilizados para diferentes juegos serios educativos. En este sentido, el objetivo principal de los investigadores consistió en identificar los componentes candidatos de los SG que podrían desarrollarse como servicios para juegos dentro de diferentes géneros y dominios.

Para identificar los componentes que se pueden reutilizar en formato de servicio, (Carvalho, Bellotti, Hu, et al., 2015) utilizan el modelo ATMSG, el cual se encarga de delinejar un modelo que representa varios componentes de bajo nivel de un juego serio a medida que este se desarrolla y la manera en cómo están conectados a los objetivos educativos y de entretenimiento de alto nivel del juego. Dicho modelo, considera que los juegos serios se utilizan en el contexto de las siguientes actividades: i) La actividad de juego, ii) La actividad de aprendizaje, iii) La actividad instruccional. Estas actividades se descomponen en una secuencia de acciones mediadas por herramientas con objetivos específicos, lo que da como

resultado un conjunto de categorías que constituyen la base de una taxonomía de componentes de juegos serios, en la que las taxonomías existentes de aprendizaje, instrucción, juegos y juegos serios se reorganizan según el modelo ATMSG (Carvalho, Bellotti, Hu, et al., 2015). Dichas actividades se pueden observar en la Figura 7.22.



**Figura 7.22 Actividades por acciones con herramientas y objetivos específicos**  
**Fuente:** Figura tomada del texto de (Carvalho, Bellotti, Hu, et al., 2015, p. 148).

Para identificar los elementos reutilizables de los juegos serios, se utilizarán los siguientes criterios de selección:

- La relevancia para la eficacia de los juegos serios.
- La posibilidad de reutilización en diferentes juegos y dominios de aprendizaje.

Una vez identificados y recopilados los componentes de los juegos serios mediante los criterios de selección, se reagruparon según sus dominios y funcionalidades, para identificar los grupos de componentes reutilizables para ser implementados como servicios en un marco SOA para juegos serios.

Los componentes con sus dominios funcionales identificados de la taxonomía ATMSG que podrían implementarse como servicios se pueden observar en la Tabla 7.19.

Asimismo, para cada uno de los dominios funcionales de la Tabla 7.19, (Carvalho, Bellotti, Hu, et al., 2015) identifican una lista de funcionalidades que podrían implementarse como servicios, a saber:

- ✓ **Interacción entre jugadores** - Servicios para recopilar, mostrar y comparar puntuaciones.
- ✓ **Interacción alumno-instructor** - Servicios para consultar a los jugadores/estudiantes, para que respondan a las preguntas.
- ✓ **Almacenamiento y recuperación de información** - Servicios relacionados con el conocimiento dentro del juego o sobre el propio dominio de aprendizaje.
- ✓ **Realimentación** - Servicios que proporcionan una manera de enviar los resultados de la evaluación de vuelta al juego, para permitir la provisión de realimentación de aprendizaje, y para apoyar la auto reflexión del jugador sobre el aprendizaje.
- ✓ **Adaptabilidad** - Servicios encargados de consolidar la información precedente de varios servicios de evaluación.
- ✓ **Conectores del juego** - Servicios que proporcionan módulos adaptadores y modelos de datos que enlazan servicios externos con el juego.
- ✓ **Perfil de usuario** - No se deriva directamente de la taxonomía, pero se requiere un servicio de perfil de usuario común para permitir la interacción, sincronización y la persistencia de las características en diferentes juegos y entornos de aprendizaje.

Una vez identificados las funcionalidades clave de los componentes reutilizables y relevantes para el desarrollo de juegos serios, los investigadores (Carvalho et al., 2015), proponen la refactorización de un juego existente, llamado “The Journey”, esto con el objetivo de definir las direcciones para su futura mejora. Según los autores, la migración de los juegos serios a una arquitectura SOA basada en los componentes propuestos, puede ayudar a los desarrolladores a estructurar

los juegos serios de una manera que promueva la reutilización y la capacidad de composición.

**Tabla 7.19 Resumen - Aspectos de análisis de (Carvalho, Bellotti, Hu, et al., 2015)**

Actividad	Tipos de elementos	Items	Dominios funcionales
Juego	Acciones	Escucha de eventos	Conectores de juegos
		Ver/Escuchar/Leer información	Almacenamiento de información y recuperación
		Preguntas	Almacenamiento de información y recuperación
	Herramientas	Puntuación de la red social	Interacción entre jugadores
		Tablas de clasificación	Interacción entre jugadores
	Objetivos	Competición	Interacción entre jugadores
Aprendizaje	Acciones	-	-
	Herramientas	Encuestas, cuestionarios	Interacción alumno - instructor
		Diario del estudiante	Interacción alumno-instructor
		Activos multimedia (audios, películas, gráficos, etc)	Almacenamiento de información y recuperación
	Objetivos	Observación reflexiva	Realimentación
		Aprender a aprender	Realimentación
Instrucción intrínseca	Acciones	Evaluación cuantitativa	Evaluación
		Andamio	Adaptabilidad
		Mostrar problemas similares	Adaptabilidad
		Apoyo a la recuperación de errores	Adaptabilidad
	Herramientas	Medidas de rendimiento	Evaluación
		Evaluar el rendimiento	Evaluación
		Proporcionar información	Realimentación
		Confianza	Evaluación, Adaptación
Instrucción extrínseca	Objetivos	Satisfacción	Evaluación, Adaptación
		Evaluación cualitativa	Evaluación
		Medidas de rendimiento	Evaluación
		Evaluar el rendimiento	Evaluación
	Objetivos	Proporcionar información	Realimentación

**Fuente:** Tabla tomada y adaptada a partir del texto de (Carvalho et al., 2015, p.149)

**Tabla 7.20 Resumen - Aspectos de análisis de (Carvalho et al., 2015)**

Categoría	Criterios	Descripción
Aspectos generales	País de la propuesta arquitectónica. Idioma de la propuesta. Nivel educativo.	Italia. Ingles. Educación secundaria y superior.
Aspectos arquitectónicos	Fundamentación. Tipo de dispositivos. Reutilización de componentes	SOA. No especifica. Si.
Aspectos pedagógicos	Destinatarios de la arquitectura. Propósito de la arquitectura. Nivel de complejidad.	Desarrolladores. Desarrollo de componentes reutilizables para juegos serios. Bajo.
Aspectos de análisis	Tipo de producto a desarrollar (juego) Resultados de utilización obtenidos. Guías o ayudas disponibles.	Juegos serios. Simplificación del proceso de desarrollo de los juegos serios. No.

#### 4.16.10. Propuesta de (Scacchi, 2017)

En **Estados Unidos**, (Scacchi, 2017) realizó un estudio con el objetivo de analizar los aspectos de la ingeniería de software (SE) contemporánea de los juegos para computadoras, de esta manera, investigó acerca de las tecnologías de desarrollo, lo cual se hizo mediante un caso práctico de aplicación de técnicas de SE a través de la utilización de juegos. Según (Scacchi, 2017), los juegos de computadora forman parte de un ámbito significativo e interesante para la investigación innovadora en técnicas y tecnologías de SE, por ello, el autor destaca la importancia de determinar los posibles intereses comunes los cuales permitan mejorar la manera de diseñar el software para juegos.

Según (Scacchi, 2017), los videojuegos podrían ser el ámbito por excelencia de la investigación y el desarrollo en informática e ingeniería de software, ya que, el desarrollo de juegos implica la integración y el equilibrio de capacidades de software procedentes de diferentes áreas, es decir, los juegos de computadora favorecen la investigación y el desarrollar nuevas formas y medios de ingeniería de software. En este sentido, los mismos investigadores agregan que, existen diferentes tipos de kits de desarrollo de software comerciales o de código abierto, motores, servicios y enfoques para producir, entregar y evolucionar juegos para computadoras de diferentes géneros, mismos que se pueden observar en la [Tabla 7.21](#).

Los juegos de computadora suelen representar configuraciones de múltiples componentes de software, bibliotecas y servicios de red, sin embargo, en el enfoque de estos tipos de juegos como medios interactivos a menudo se considera que la arquitectura de software es poco o nada relevante para el diseño de juegos, de tal manera que crea una brecha entre los juegos de computadora (servidor único o en entorno de ejecución de juegos para pc) y los juegos en los que se deben proporcionar servicios distribuidos, donde la arquitectura del sistema es fundamental.

Es importante resaltar los cuatro tipos de arquitecturas de software que se suelen utilizar para el diseño de juegos multijugador en red (Scacchi, 2017):

- ✓ Arquitectura de tiempo de ejecución estáticas y dinámicas para un motor de juego.
- ✓ Arquitectura de los marcos de desarrollo de juegos o kits de desarrollo de software que integran la arquitectura de desarrollo de un juego junto con su motor de juego.
- ✓ Distribución arquitectónica de la funcionalidad del software y los servicios de procesamiento de datos para los juegos.
- ✓ Arquitectura informativa y geográfica de los niveles de juego como espacios de juego diseñados.

**Tabla 7.21** *Tecnologías para la ingeniería de software de juegos para computadora*

Juego SDK* o motor de juegos	Ejemplos comerciales	Características de desarrollo del juego	Alternativas de software de código abierto	Plataformas de desarrollo o de destino	Géneros comunes de juegos
HTML5 o web	Construct 2, GameSalad	Basado en reglas, UI procesamiento de eventos	EaselJS, GDevelop, Kiwi.js, Phaser	PC o dispositivos con navegadores	2D basados en web
Genero de juego específico	Adventure Game Studio, Minecraft, RPG Maker, SAGE	Interfaz de usuario basada en el género, experiencia de usuario	Freeciv, Minetest, Ren'Py, Quest, Stratagus	PC en red	Aventura y de rol, estrategia en tiempo real, novelas visuales
Biblioteca, marco de	GameMaker,	Primitivas de programación de	ANX, Cocos2d, OGRE	PC	Juegos 2D o 3D para un solo

## CAPÍTULO VIII. PROPUESTA METODOLÓGICA Y ARQUITECTÓNICA PARA EL DISEÑO DE JS

Juego SDK* o motor de juegos	Ejemplos comerciales	Características de desarrollo del juego	Alternativas de software de código abierto	Plataformas de desarrollo o de destino	Géneros comunes de juegos
trabajo o entorno de ejecución	libGDX, Microsoft, XNA	juegos, API abiertas			usuario o para varios
Modificación del juego	Half-Life, Neverwinter Nights, Unreal	Modificación o reutilización de juegos en funcionamiento	Doom, Quake, Quake Arena	PC en red	Depende del juego de origen
Juego de IDE	CryEngine, Source, Unity, Unreal Engine, UDK	Flujo de trabajo de calidad de producción	Blender, Torque 3D	PC	Juegos de masas, juegos de acción y disparos en 3D en primera persona
Servicio basado en la nube o *MMOG	Amazon Lumberyard, Facebook, Steam, Twitch	Servicios escalables y comercio electrónico seguro	OpenSimulator, Worldforge	PCs, consolas, conectados a Internet smartphones	eSports, juegos free to play, MMOGs

\*SDK significa kit de desarrollo de software; MMOG se refiere a juego multijugador masivo en línea.

**Fuente:** Tabla tomada y adaptada a partir del texto de (Scacchi, 2017, p. 111)

Para el caso de estudio, (Scacchi, 2017) utilizó el juego llamado “*The Beam Game*”, el cual fue creado para ayudar a comprender la física de los rayos. El juego se llevó a cabo de forma ágil e incremental, de modo que ayudó a identificar los requisitos funcionales o no funcionales. Igualmente, se utilizó un marco arquitectónico de sistema basado en reglas y dirigido por eventos, que apoya la creación rápida de prototipos de medios interactivos o juegos para su despliegue con navegadores web.

**Tabla 7.22 Resumen - Aspectos de análisis de (Scacchi, 2017)**

Categoría	Criterios	Descripción
Aspectos generales	País de la propuesta arquitectónica. Idioma de la propuesta. Nivel educativo.	USA. Inglés. Todos.
Aspectos arquitectónicos	Fundamentación. Tipo de dispositivos. Reutilización de componentes	Ingeniería de software (SE). Escritorio y web. Si.
Aspectos pedagógicos	Destinatarios de la arquitectura. Propósito de la arquitectura. Nivel de complejidad.	Desarrolladores. Desarrollo. Bajo.
Aspectos de análisis	Tipo de producto a desarrollar (juego) Resultados de utilización obtenidos. Guías o ayudas disponibles.	Juego serio. No. No.

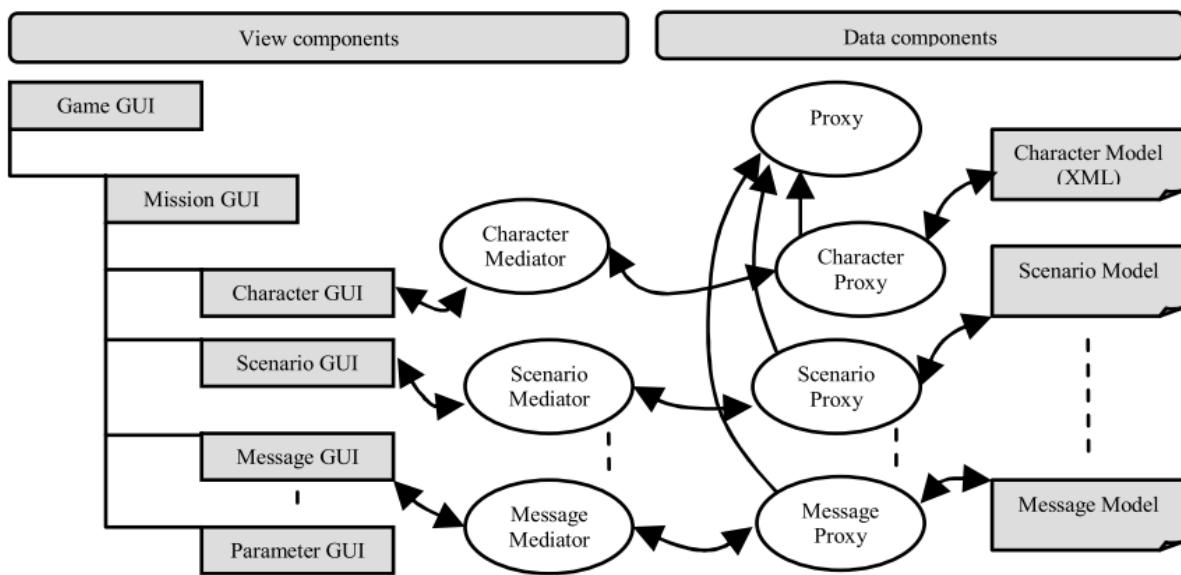
### 4.16.11. Propuesta de (Yessad et al., 2010)

En **Francia**, los investigadores (Yessad et al., 2010) indican que los videojuegos se han identificado como posibles herramientas de aprendizaje, los cuales favorecen alcanzar objetivos pedagógicos (juegos serios), los mismos investigadores afirman que para lograr incluir los objetivos pedagógicos específicos en un juego, se deben modificar ciertas particularidades del juego; es decir, dependiendo de las características que posea o no, así varían los objetivos que permite alcanzar. En este sentido, la posibilidad de modificar estas características va a depender en gran medida de la flexibilidad del motor de juego (Yessad et al., 2010).

Por la razón anterior, (Yessad et al., 2010) presentan un Entorno de Creación de Juegos Serios (SeGAE, *Serious Game Authoring Environment*), el cual consiste en un entorno de desarrollo amigable con los desarrolladores de video-juegos, además, permite modificar diferentes aspectos del diseño del juego: i) Avatares, Características de los personajes, Misiones y objetivos, Condiciones de victoria y, Acciones autorizadas.

SeGAE garantiza la coherencia entre estas características del juego. Para garantizar su funcionamiento, los investigadores utilizaron un juego serio llamado *Blossom Flowers*, orientado a personas en el área empresarial (Yessad et al., 2010). Para integrar el entorno de autoría con el juego serio, se utilizó un esquema de definición XML para describir personajes, eventos, acciones y otros elementos básicos del juego. Posteriormente, los desarrolladores se encargan de transformar los objetivos pedagógicos en objetivos de juego en sí e integrarlos en los distintos niveles del juego. En este sentido, SeGAE permite configurar y adaptar el juego en tiempo real de acuerdo con las necesidades pedagógicas, mediante el uso de una descripción en XML (Yessad et al., 2010).

Para la **implementación** de SeGAE, se utilizó *Flex Builder*, el cual es un IDE construido en la plataforma Eclipse, para la creación de *Rich Internet Applications* (RIAs). Cabe destacar que, para su diseño se utilizó la arquitectura *Model-View-Controller* (MVC). Como se mencionó anteriormente, los modelos a utilizar se crean en XML, MXML y en lenguaje ActionScript (ver [Figura 7.23](#)).



**Figura 7.23 Arquitectura SeGAE**

Fuente: Figura tomada del texto de (Yessad et al., 2010, p. 3).

El entorno SeGAE implementa diferentes editores (Yessad et al., 2010):

- ✓ **Editor de personaje** - Permite editar las características espaciales y visuales del avatar del jugador y los diferentes personajes de la misión. Cabe destacar que la posición espacial de personajes presentes en el juego es elegida de una matriz de coordenadas especiales autorizadas en la escena.
- ✓ **Editor de escenario temporal** - Permite especificar la máquina finita de estados que modela la secuencia de las acciones del juego. El instructor debe definir los eventos en la misión y para cada uno definir las pre-condiciones y post-condiciones. El orden de aparición de eventos define el escenario de la misión. Algunos eventos pueden aparecer simultáneamente mientras que otros aparecen en secuencia.
- ✓ **Editor de mensaje y diálogo** - Permite agregar mensajes y diálogos en las misiones del juego. SeGAE toma en cuenta diferentes lenguajes de usuario. Dichos mensajes son utilizados por el instructor para la edición de escenarios temporales como, por ejemplo, mostrar un mensaje al jugador.

- ✓ **Internacionalización del juego** - Permite la adaptación de lenguaje de los mensajes y diálogos del juego al usuario final.

SeGAE ofrece la posibilidad de modificar y adaptar un juego serio a necesidades pedagógicas específicas mediante la definición de nuevas misiones y objetivos en tiempo real. Este entorno interopera con el juego serio mediante archivos XML.

**Tabla 7.23 Resumen - Aspectos de análisis de (Yessad et al., 2010)**

Categoría	Criterios	Descripción
Aspectos generales	País de la propuesta arquitectónica. Idioma de la propuesta. Nivel educativo.	Francia. Inglés. Todos.
Aspectos arquitectónicos	Fundamentación. Tipo de dispositivos. Reutilización de componentes	Model View Controller (MVC). Web. Sí.
Aspectos pedagógicos	Destinatarios de la arquitectura. Propósito de la arquitectura. Nivel de complejidad.	Docentes. Propiciar la secuencialidad del aprendizaje. Bajo.
Aspectos de análisis	Tipo de producto a desarrollar (juego) Resultados de utilización obtenidos. Guías o ayudas disponibles.	Juegos serios. No. No.

#### 4.16.12. Propuesta de (Costa et al., 2016)

En **Brasil**, los investigadores (Costa et al., 2016) señalan que tanto los juegos serios (JS) y los entornos virtuales (EV) han sido utilizados para distintos propósitos, particularmente, en el área de la salud; los cuales proveen un balance entre el involucramiento y el aprendizaje, potenciando el desarrollo de conocimientos y habilidades. Por ello, (Costa et al., 2016) realizaron una investigación vinculada con las capacidades de los juegos serios y de los entornos virtuales para favorecer la adquisición de habilidades y competencias profesionales. Además, los investigadores presentan una propuesta de una arquitectura orientada a portales, la cual es capaz de asistir en el planeamiento y la evaluación del uso de JS y EV en el ámbito de entrenamiento en el área de la salud. La propuesta involucra la posibilidad de uso de diferentes juegos y simulaciones en secuencia para actividades de entrenamiento (Costa et al., 2016).

A partir de la investigación realizada, (Costa et al., 2016) identificaron aspectos importantes por ser considerados en la propuesta que realizan:

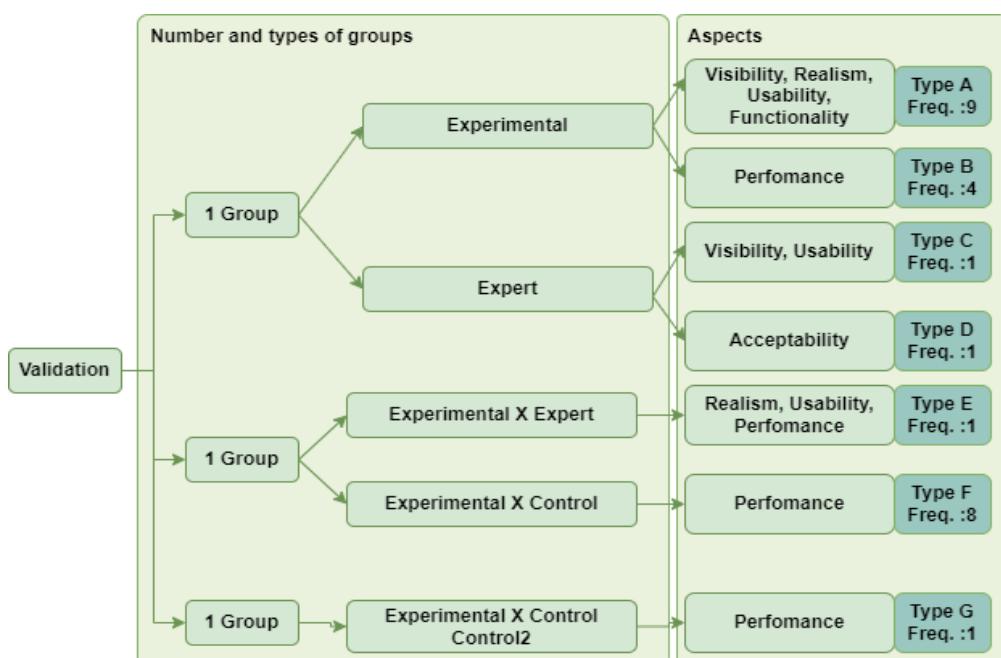
✓ **Evaluación del usuario** - Este aspecto se considera importante en los

juegos serios orientados a entrenamiento, en cuanto al desarrollo y de habilidades y conocimiento, debido a que obtuvieron como resultado que el 66% de los trabajos analizados, se utilizan sistemas de evaluación automática.

✓ **Validación de JS y EV** - Este aspecto se considera importante debido a que

entre los trabajos analizados se observaron algunas diferencias, tales como: la cantidad de grupos utilizados para la validación, el tipo de usuarios por grupo y algunos aspectos validados en el experimento.

En general, independientemente del método utilizado para realizar la validación, los resultados son positivos en cuanto al entrenamiento de personas. En la [Figura 7.24](#) se pueden observar con mayor detalle los procesos de validación identificados en las distintas investigaciones analizadas por (Costa et al., 2016).



**Figura 7.24 Procesos de validación identificados en las investigaciones**

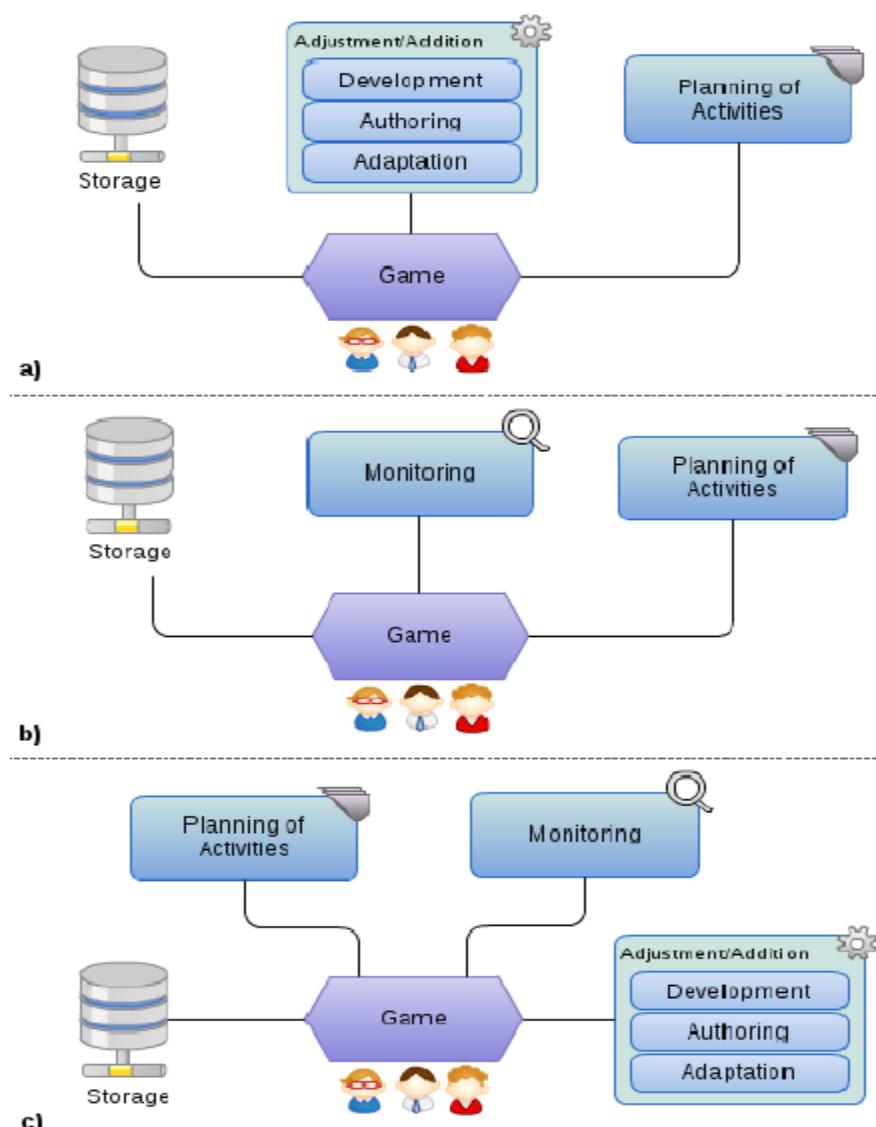
Fuente: Elaboración propia a partir del texto de (Costa et al., 2016, p. 3)

Igualmente, (Costa et al., 2016) analizaron una serie plataformas o portales, entre ellas: Pingo, AVEP, Simurena, mGBL, entre otros, dicho análisis les permitió identificar cuatro servicios principales que se integran en dichas plataformas:

- ✓ Almacenamiento.
- ✓ Ajuste o adición.
- ✓ Planeamiento de actividades.
- ✓ Monitoreo.

Estos servicios se pueden combinar de distintas maneras según sea el caso. En la [Figura 7.25](#) se pueden apreciar algunos de ellos.

Con respecto al **planeamiento de actividades**, cabe destacar que existen las posibilidades de planeamiento horizontal, vertical e híbrido. Por otro lado, para el **monitoreo**, es importante aclarar que involucra el uso de procesos de *machine decision* para asistir la verificación del desempeño del usuario (Costa et al., 2016).



**Figura 7.25 Ajustes de los servicios brindados por las plataformas o portales**

**Nota:** (a): Almacenamiento, ajuste o adición, y planeamiento de actividades; (b): Almacenamiento, monitoreo y planeamiento de actividades; (c): unión de los servicios: almacenamiento, ajuste o adición, planeamiento de actividades y monitoreo.

**Fuente:** Figura tomada del texto de (Costa et al., 2016, p. 4)

En esta línea, la arquitectura propuesta por (Costa et al., 2016) representa un esquema general para portales web que permite proveer un conjunto de servicios y mecanismos que pueden ser usados por el mediador para organizar y evaluar secuencias de actividades educativas que involucren JS y EV. Para lograr este propósito, (Costa et al., 2016) se plantean diferentes supuestos:

- ✓ Estudiantes y profesionales pueden verificar su aptitud a través de JS y EV.

- ✓ Los JS y EV relacionados con apoyo educativo están enfocados en cuestiones específicas y tienen metas educativas propuestas para distintas dimensiones del conocimiento.
- ✓ Los mediadores son los responsables de guiar el proceso educativo, y pueden organizar conjuntos secuenciales de actividades como caminos de aprendizaje.

Para alcanzar dichos supuestos, los esfuerzos que se realicen en la definición de la arquitectura deberían seguir las siguientes pautas:

- ✓ Servir como apoyo educativo, permitiendo la evaluación del desempeño durante las actividades secuenciales.
- ✓ Permitir a los mediadores crear secuencias de actividades, y su respectiva evaluación, con base en los objetivos de aprendizaje.
- ✓ Permitir a estudiantes realizar secuencias de actividades y evaluar su desempeño.
- ✓ Permitir a los mediadores llevar un registro del desempeño de los estudiantes.
- ✓ Permitir a los mediadores la solicitud de inclusión de nuevos juegos o entornos virtuales al sistema.
- ✓ Permitir el acceso vía web.

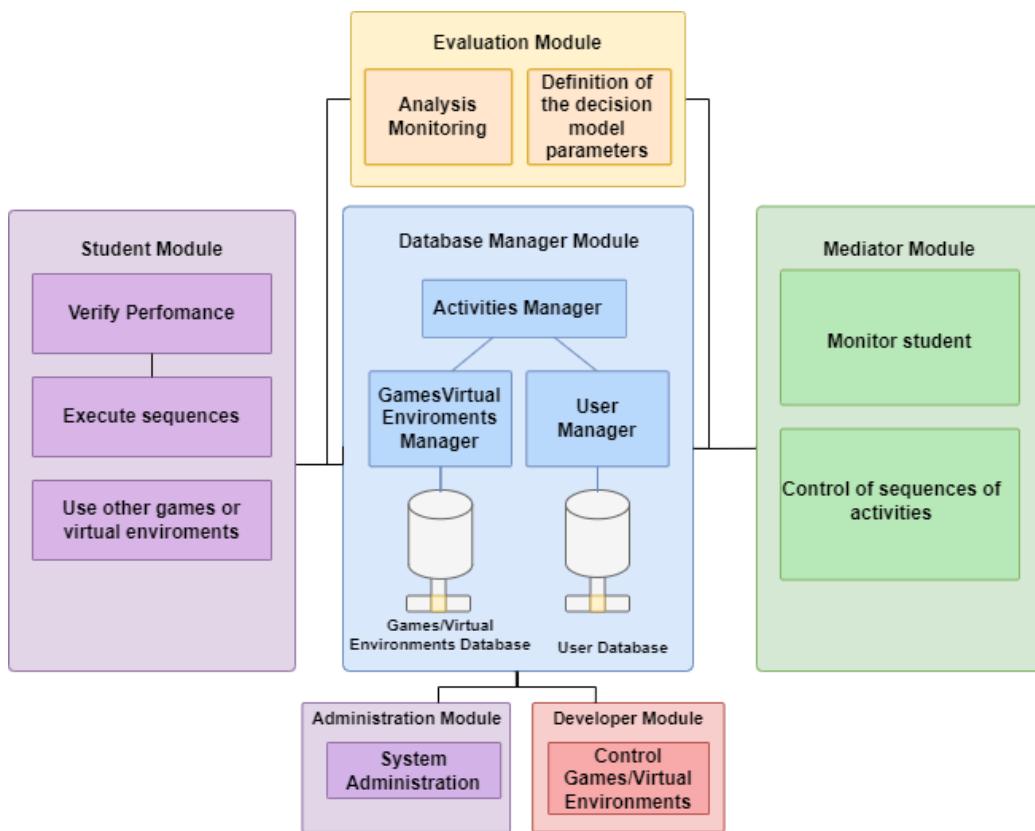
En este contexto, la secuencialidad consiste en la organización de tareas y contenidos planeados por el mediador de acuerdo con los objetivos de aprendizaje que se desean alcanzar. Las actividades que conforman una secuencia corresponden a los juegos serios y entornos virtuales que pueden ser seleccionados y organizados en niveles de complejidad (Costa et al., 2016).

Los niveles son jerárquicos, y cada uno está conformado por un número de juegos o entorno. La progresión entre niveles se alcanza de acuerdo con los requerimientos mínimos del nivel actual; no obstante, los JS o EV agrupados en un mismo nivel no tienen restricciones en el orden de realización (Costa et al., 2016).

Los requerimientos mínimos para avanzar entre niveles serán definidas por los mediadores, y la secuenciación debe ser organizada y guiada con el objetivo de proporcionar aprendizajes específicos a través de la experimentación y el entrenamiento, así como ofrecer retos con distintos niveles de dificultad (Costa et al., 2016). La arquitectura propuesta por (Costa et al., 2016) fue dividida en tres capas:

- ✓ Presentación.
- ✓ Actividades y mecanismos de evaluación.
- ✓ Administración.

En la [Figura 7.26](#) se puede observar con mayor detalle el diagrama general de la arquitectura, en donde se muestran los seis módulos: evaluación, manejo de bases de datos, administración, desarrollador, mediador y estudiante.



**Figura 7.26 Arquitectura general propuesta por (Costa et al., 2016)**

**Fuente:** Elaboración propia a partir del texto de (Costa et al., 2016, p. 7)

Finalmente, (Costa et al., 2016) indican que, en el momento de proponer una arquitectura que integre JS y EV se debería considerar que:

- ✓ Es fundamental realizar la evaluación del desempeño del estudiante y, se debería tomar en cuenta en la creación de la arquitectura.
- ✓ Es importante realizar la verificación del alcance de las metas propuestas en las aplicaciones creadas.
- ✓ Es importante que los proyectos incluyan una etapa de validación.
- ✓ Es importante incorporar la secuenciación, ya que permite analizar en distintos niveles las habilidades cognitivas, afectivas y psicomotoras.

Tabla 7.24 Resumen - Aspectos de análisis de (Costa et al., 2016)

Categoría	Criterios	Descripción
Aspectos generales	País de la propuesta arquitectónica. Idioma de la propuesta. Nivel educativo.	Brasil. Inglés. Todos.
Aspectos arquitectónicos	Fundamentación. Tipo de dispositivos. Reutilización de componentes	*Pingo, GBL. Escritorio y web. Sí.
Aspectos pedagógicos	Destinatarios de la arquitectura. Propósito de la arquitectura. Nivel de complejidad.	Desarrolladores. Potenciar el aprendizaje. Medio.
Aspectos de análisis	Tipo de producto a desarrollar (juego) Resultados de utilización obtenidos. Guías o ayudas disponibles.	Entornos para juegos serios. No. Sí, Infografías y ficha técnica

\***Pingo** (Portal of Instructional Games, Online). **mGBL** (Mobile Game Based Learning).

#### 4.16.13. Propuesta de (Ismail & Belkhouche, 2019)

En **Emiratos Árabes Unidos**, los investigadores (Ismail & Belkhouche, 2019) señalan que diferentes psicólogos y expertos en educación, han enfatizado durante varios años la importancia de la personalización del aprendizaje para una educación más efectiva, misma mediada o complementada a través de diferentes herramientas tecnológicas. Es decir, un aprendizaje personalizado se refiere a que el ritmo de enseñanza debe adecuarse a las necesidades de aprendizaje y facilitarse según preferencias e intereses específicos de la persona que aprende (Ismail & Belkhouche, 2019).

En este sentido, los sistemas de software para el aprendizaje personalizado proveen experiencias únicas de aprendizaje, tomando en cuenta aspectos tales como el contexto, los niveles de destreza y los estilos de aprendizaje, donde estos sistemas cubren un amplio espectro de posibilidades de personalización, misma que dificulta en ocasiones la comunicación entre el usuario y los diseñadores técnicos que crean estos sistemas, por ello, es de suma importancia contar con arquitecturas de software que faciliten dicha comunicación entre investigadores (Ismail & Belkhouche, 2019).

Los investigadores (Ismail & Belkhouche, 2019) se propusieron como objetivo plantear una arquitectura de software reutilizable para la creación de estos sistemas, misma que requiere ser creada con un nivel de abstracción que facilite el trabajo en múltiples contextos.

Ante este contexto tecnológico, los investigadores consideran que es de suma importancia tener claro los siguientes conceptos (Ismail & Belkhouche, 2019):

- ✓ **Sistemas de software para la personalización del aprendizaje** - Sistemas que adaptan el acceso a los recursos de aprendizaje, dentro de un ambiente de software y a un modelo de usuario.
- ✓ **Ambiente de aprendizaje en software** - Sistemas caracterizados por el tipo de tecnología utilizado para su implementación (móvil, web, PC.). Cada tipo conlleva distintos atributos y rasgos de personalización.
- ✓ **Recursos de aprendizaje** - Componentes como cursos en línea, libros electrónicos, actividades evaluativas.
- ✓ **Modelos del usuario** - Procesos de inferencia de información acerca de las personas en etapa de aprendizaje, mediante características, preferencias o comportamiento; se puede crear de manera automatizada o manual. Al crear un modelo, es importante preguntarse: ¿Cuáles aspectos de la persona se necesitan modelar?, ¿Cuáles datos se pueden utilizar para inferir el modelo requerido?, ¿Cómo se recolectará la información?, y finalmente ¿Cómo se creará el modelo?

La arquitectura propuesta por (Ismail & Belkhouche, 2019) para los sistemas de software orientados a la personalización del aprendizaje, posee cuatro componentes importantes:

- ✓ **Unidad del aprendiz** - Se encarga de mantener la información relevante del usuario, como preferencias, destrezas, intereses o conocimientos.
- ✓ **Unidad de conocimiento** - Se encarga de mantener los recursos de aprendizaje.

- ✓ **Unidad de personalización** - Es el componente en donde se crea el modelo del aprendiz y se mapea con los recursos de aprendizaje, mediante una lógica de personalización.
- ✓ **Unidad de presentación** - Se encarga de representar el entorno de software mediante el cual se dará la interacción con el usuario.

Estos componentes llevan a cabo varias funciones, mismas que se detallan en la Tabla 7.25.

**Tabla 7.25 Funciones del sistema de aprendizaje personalizado**

Función	Descripción
Mantener data del aprendiz	Monitorear el comportamiento actual del aprendiz, leer data histórica como por ejemplo el historial de búsqueda, <i>click logs</i> , clasificar data o recolectando inputs explícitos del aprendiz.
Personalizar los recursos de aprendizaje	Analizar la data recopilada del usuario, construir un modelo representativo del aprendiz y mapear los recursos de aprendizaje basado en el modelo del aprendiz.
Mantener recursos de aprendizaje	Almacenar, organizar y actualizar los recursos de aprendizaje.
Presentar los recursos de aprendizaje	Presentar recursos de aprendizaje personalizados.

**Fuente:** Tabla tomada y adaptada a partir del texto de (Ismail & Belkhouche, 2019, p. 3)

Segundo, los investigadores (Ismail & Belkhouche, 2019) realizan un “*Mapeo arquitectónico entre elementos de los sistemas de software para la personalización del aprendizaje y el modelo propuesto*” en donde “tipifican” múltiples trabajos de investigación pertenecientes al área de personalización del aprendizaje, en distintos contextos, donde se realiza un mapeo de los componentes de estos sistemas acorde a las cuatro unidades arquitectónicas propuestas anteriormente (aprendiz, conocimiento, personalización, presentación).

Los resultados se resumen en la Tabla 7.26, los cuales reúnen los aportes de diferentes investigadores de referencia en el área de estudio (Bhatia & Prasad, 2015; Biletskiy et al., 2009; Ismail, 2018; Khribi et al., 2008; Pereira et al., 2012).

**Tabla 7.26 Mapeo de sistemas de software de aprendizaje personalizado**

Sistema de personalización	Unidad del aprendiz	Unidad de conocimiento	Unidad de personalización	Unidad de presentación
(Bhatia & Prasad, 2015)	Explícito: Clasificación histórica de data	e-books	Data mining-based approach	Curso en línea

**CAPÍTULO VIII. PROPUESTA METODOLÓGICA Y ARQUITECTÓNICA PARA EL DISEÑO DE JS**

Sistema de personalización	Unidad del aprendiz	Unidad de conocimiento	Unidad de personalización	Unidad de presentación
(Biletskiy et al., 2009)	Explícito: Perfil del aprendiz IMS	Objetos de aprendizaje	Rule-based	Curso en línea
(Khribi et al., 2008)	Implícito: Uso histórico de la web y sesiones de interacción activas en la web	Objetos de aprendizaje	Data mining-based & Information Retrieval-based	Curso en línea
(Pereira et al., 2012)	Implícito: Interacciones actuales del usuario	Vocabulario, imágenes y bases de datos de funciones o tareas	Data mining-based approach	Juego
(Ismail, 2018)	Implícito: Comportamiento sobre la navegación	Texto desestructurado	Information Retrieval-based	Wiki

**Fuente:** Tabla tomada y adaptada a partir del texto de (Ismail & Belkhouche, 2019, p. 4)

En la actualidad la creación de sistemas de software para la personalización del aprendizaje, presenta complicaciones debido a la dificultad de comunicación existente entre desarrolladores y usuarios (Ismail & Belkhouche, 2019).

En resumen, los investigadores (Ismail & Belkhouche, 2019) elaboraron una arquitectura de software para la creación de piezas de software de una manera más fácil, además, realizaron una definición general de conceptos que facilitan la comunicación entre usuarios y diseñadores. Por último, se realizó un mapeo de distintos sistemas creados con la arquitectura propuesta, para verificar su utilidad en contextos variados.

**Tabla 7.27 Resumen - Aspectos de análisis de (Ismail & Belkhouche, 2019)**

Categoría	Criterios	Descripción
Aspectos generales	País de la propuesta arquitectónica. Idioma de la propuesta. Nivel educativo.	Emiratos Árabes Unidos. Inglés. Todo nivel.
Aspectos arquitectónicos	Fundamentación. Tipo de dispositivos. Reutilización de componentes	Aprendizaje personalizado. Móvil, web. Sí.
Aspectos pedagógicos	Destinatarios de la arquitectura. Propósito de la arquitectura. Nivel de complejidad.	Desarrolladores, docentes. Potenciar el aprendizaje personalizado. Medio.
Aspectos de análisis	Tipo de producto a desarrollar (juego) Resultados de utilización obtenidos. Guías o ayudas disponibles.	Sistemas de aprendizaje personalizado. No. Sí, infografías.

En la [sección 7.4](#), se abordarán resultados obtenidos al aplicar los criterios de análisis a las 13 arquitecturas descritas anteriormente y seleccionadas para el estudio.

## 4.17. Análisis de resultados según criterios de evaluación

Este apartado muestra los principales resultados obtenidos a partir de la aplicación de los criterios de análisis a las propuestas de arquitecturas o investigaciones seleccionadas, las cuales han sido descritas en la sección 7.3.

### 4.17.1. Aspectos generales

En relación con el **País** donde se desarrolla la propuesta arquitectónica o investigación, se puede observar en detalle en la [Tabla 7.28](#) que en Italia, Brasil, Francia y México son los países que contabiliza un mayor número de propuestas con 2 (15%) cada uno, y todos los demás con 1 propuesta respectivamente (8% cada uno), esto podría significar que, en el continente americano se están dando los primeros pasos en el diseño de propuestas arquitectónicas para el desarrollo de juegos serios.

**Tabla 7.28 Criterio - País donde se desarrolla la propuesta arquitectónica**

Autor/es	Idioma	País
(Gutiérrez-Hernández et al., 2013)	Español	México
(Flores et al., 2019)	Español	México
(Maggiorini et al., 2016)	Inglés	Italia
(Carvalho, Bellotti, Hu, et al., 2015)	Inglés	Italia
(El Mawas, 2014)	Inglés	Francia
(Söbke & Streicher, 2016)	Inglés	Alemania
(Stavrev et al., 2018)	Inglés	República Checa
(Mizutani et al., 2021)	Inglés	Brasil
(Ollsson et al., 2015)	Inglés	Suecia
(Scacchi, 2017)	Inglés	Estados Unidos
(Yessad et al., 2010)	Inglés	Francia
(Costa et al., 2016)	Inglés	Brasil
(Ismail & Belkhouche, 2019)	Inglés	Emiratos Árabes Unidos

Fuente: Elaboración propia, 2021

En relación con el criterio de **idioma** de la propuesta arquitectónica o investigación, se puede visualizar en la [Tabla 7.28](#) que 11 (85%) de las propuestas seleccionadas para este estudio, han sido escritas en inglés y, 2 (15%) en idioma español, estos datos podrían indicar que los investigadores buscan dar a conocer los resultados de sus investigaciones académico-científicas en un idioma universal, el cual les permita visualizarlas e impactar a una mayor cantidad

de lectores o investigadores a nivel internacional, para este caso en específico, inglés.

Respecto al criterio **nivel educativo**, la [Tabla 7.29](#) resume la información de las propuestas arquitectónicas analizadas, evidenciando si la propuesta fue enfocada o destinada para una población en específico, o bien, para varias de ellas.

Tal como se puede observar en la [Tabla 7.29](#) la mayoría de las propuestas arquitectónicas analizadas han sido pensadas para la educación superior/universitaria con 9 (69.23%) de las 13 propuestas, seguido por el nivel primario con 6 (46.15%) de las 13 arquitecturas analizadas. En esta línea, las propuestas realizadas por (Costa et al., 2016; Ismail & Belkhouche, 2019; Scacchi, 2017; Yessad et al., 2010) son las que se identificaron que fueron pensadas para ser utilizadas en todos los niveles educativos. Por último, las propuestas de (Mizutani et al., 2021; Ollsson et al., 2015) fueron enfocadas directamente a la industria y, no se infiere algún tipo de nivel educativo en específico.

**Tabla 7.29** Criterio - Nivel educativo según propuestas arquitectónicas

Metodología	Nivel educativo						
	Especial	Inicial	Primario	Secundario	Terciario	Universitario	Otro
(Maggiorini et al., 2016)						✓	
(El Mawas, 2014)						✓	
(Gutiérrez-Hernández et al., 2013)			✓				
(Söbke & Streicher, 2016)						✓	
(Stavrev et al., 2018)						✓	
(Flores et al., 2019)			✓				
(Mizutani et al., 2021)							Industria
(Ollsson et al., 2015)							Industria
(Carvalho, Bellotti, Hu, et al., 2015)				✓		✓	
(Scacchi, 2017)	✓	✓	✓	✓	✓	✓	
(Yessad et al., 2010)	✓	✓	✓	✓	✓	✓	
(Costa et al., 2016)	✓	✓	✓	✓	✓	✓	
(Ismail & Belkhouche, 2019)	✓	✓	✓	✓	✓	✓	

Los resultados de la [Tabla 7.29](#) podrían indicar que a nivel general existe una tendencia a que las propuestas arquitecturas para el diseño o desarrollo de juegos

serios, se estén orientando más al nivel de educación superior/universitario, quizás por la variedad de áreas del conocimiento en las que podrían ser aplicados.

#### 4.17.2. Aspectos arquitectónicos

En relación con la aplicación del criterio **fundamentación**, el cual permite identificar el fundamento o base que los investigadores han tomado de referencia para el diseño o propuesta de la arquitectura. Los resultados se pueden observar con detalle en la [Tabla 7.30](#).

La [Tabla 7.30](#) permite apreciar que 3 de las propuestas (Carvalho, Bellotti, Hu, et al., 2015; Söbke & Streicher, 2016; Stavrev et al., 2018) han sido fundamentadas en la arquitectura SOA, seguida por 2 en MVC (Ollsson et al., 2015; Yessad et al., 2010) y de igual forma 2 en SE (Carvalho, Bellotti, Hu, et al., 2015; Scacchi, 2017). Luego, en menor escala, las demás propuestas han sido fundamentadas en otras arquitecturas. Es importante rescatar que la propuesta de (Carvalho, Bellotti, Hu, et al., 2015) se fundamentó tomando en cuenta dos arquitecturas (SOA y SEW).

**Tabla 7.30 Criterio - Fundamentación de la arquitectura**

Propuesta arquitectónica	Fundamentación de la arquitectura*			
	SOA	MVC	SE	Otros**
(Maggiorini et al., 2016)				Microkernel
(El Mawas, 2014)				ARGILE
(Gutiérrez-Hernández et al., 2013)				ArJuS
(Söbke & Streicher, 2016)	√			
(Stavrev et al., 2018)	√			
(Flores et al., 2019)				Suave
(Mizutani et al., 2021)				Mecánica
(Ollsson et al., 2015)		√		
(Carvalho, Bellotti, Hu, et al., 2015)	√		√	
(Scacchi, 2017)			√	
(Yessad et al., 2010)		√		
(Costa et al., 2016)				Pingo
(Ismail & Belkhouche, 2019)				Aprendizaje personalizado

\*Se toman como referencia las fundamentaciones que registran 2 o más propuestas.

\*\*Modelo-Vista-Controlador (**MVC**, *Model-View-Controller*), Arquitectura orientada a servicios (**SOA**, *Service-Oriented Architectures*). Ingeniería de software (**SE**, *software engineering*). Portal de juegos didácticos en línea (**PINGO**, *Portal of Instructional Games, Online*)

A nivel general, los resultados obtenidos en la [Tabla 7.30](#), indican que actualmente se carece de una propuesta metodológica o arquitectónica sólida, que pueda servir de base o referencia como diseño arquitectónico para el desarrollo de juegos serios.

Con relación al criterio **tipo de dispositivo al que está orientada la arquitectura** (el cual busca identificar el tipo de dispositivo al que está orientada esta, es decir, si está dirigida a ser utilizada como aplicación móvil, aplicación de escritorio o aplicación web), la Tabla 7.31 presenta el resumen de los resultados obtenidos.

**Tabla 7.31 Criterio - Tipo de dispositivo al que está orientada la arquitectura**

Propuesta arquitectónica	Tipo de dispositivo al que está orientada la arquitectura		
	Escritorio	Móvil	Web
(Maggiorini et al., 2016)	✓	✓	✓
(El Mawas, 2014)			✓
(Gutiérrez-Hernández et al., 2013)			✓
(Söbke & Streicher, 2016)	✓	✓	✓
(Stavrev et al., 2018)			✓
(Flores et al., 2019)	✓		
(Mizutani et al., 2021)	✓	✓	
(Ollsson et al., 2015)	✓	✓	✓
(Carvalho, Bellotti, Hu, et al., 2015)			✓
(Scacchi, 2017)	✓		✓
(Yessad et al., 2010)			✓
(Costa et al., 2016)	✓		✓
(Ismail & Belkhouche, 2019)		✓	✓

Con base en los datos de la Tabla 7.31, se puede observar que 11 propuestas (84.62%) han sido pensadas para ser ejecutadas como aplicaciones web y 7 (53.85%) de ellas como aplicaciones de escritorio y solo 5 (38.46%) se registran como pensadas para aplicaciones móviles. Además, 3 propuestas (Maggiorini et al., 2016; Ollsson et al., 2015; Söbke & Streicher, 2016) coinciden en que han sido diseñadas para ser ofrecida en las 3 opciones de aplicación (escritorio, móvil y web). Otro aspecto importante por considerar es que, 8 propuestas (61.54%) consideran importante que la arquitectura debería ser pensada para ser ejecutada en al menos 2 tipos de aplicaciones. En suma, los resultados obtenidos dan luz en términos de que a futuro se podría pensar en una propuesta arquitectónica orientada a mínimo 2 tipos de aplicaciones, que acorde a los resultados obtenidos podrían ser web y escritorio.

Con base en el criterio de **reutilización de componentes**, la Figura 7.27 presenta los resultados obtenidos.

Sí	No
<ul style="list-style-type: none"> <li>•(Maggiorini et al., 2016)</li> <li>•(Söbke &amp; Streicher, 2016)</li> <li>•(Stavrev et al., 2018)</li> <li>•(Mizutani et al., 2021)</li> <li>•(Ollsson et al., 2015)</li> <li>•(Carvalho, et al., 2015)</li> <li>•(Scacchi, 2017)</li> <li>•(Yessad et al., 2010)</li> <li>•(Costa et al., 2016)</li> <li>•(Ismail &amp; Belkhouche, 2019)</li> </ul>	<ul style="list-style-type: none"> <li>•(El Mawas, 2014)</li> <li>•(Gutiérrez-Hernández et al., 2013)</li> <li>•(Flores et al., 2019)</li> </ul>

**Figura 7.27 Criterio - Reutilización de componentes**

Conforme a los datos de la Figura 7.27, se observa que la reutilización de componentes ha sido considerada en 10 de las propuestas analizadas (77% aproximadamente), lo cual coincide con la información presentada en la Tabla 7.31 en relación al tipo de aplicación, ya que según lo indicado por (Ralph & Wand, 2007; Sommerville, 2016), la reutilización de componentes es una característica presente en las aplicaciones web, debido a que la misma se podría dar en diferentes niveles (nivel de abstracción, nivel del objeto, nivel del componente y nivel del sistema).

#### 4.17.3. Aspectos pedagógicos

Acorde con el criterio **definición de destinatarios de la arquitectura**, el cual busca identificar al público meta para la cual fue desarrollada, se puede observar en la Tabla 7.32, que 11 propuestas (85% aproximadamente) fueron pensadas para ser utilizadas por el equipo técnico, es decir, por parte de diseñadores, desarrolladores o programadores. Luego, 7 de las propuestas (54% aproximadamente) han sido desarrolladas para que sean utilizadas por parte del personal docente, finalmente, 3 de las propuestas incluyen a la población estudiantil como destinatarios de la arquitectura. Estos resultados podrían indicar que, a nivel general las propuestas que han sido desarrolladas para el diseño de juegos serios se han orientado al equipo técnico, esto podría ser por las habilidades, la formación técnica y académica de los mismos, así como por su experticia para realizar alguna implementación adicional que pudiese ser requerida al diseñar el juego serio.

**Tabla 7.32 Criterio - Destinatarios de la arquitectura**

Metodología	Destinatarios de la arquitectura		
	Estudiantes	Docentes	Equipo técnico*
(Maggiorini et al., 2016)			✓
(El Mawas, 2014)	✓	✓	
(Gutiérrez-Hernández et al., 2013)		✓	✓
(Söbke & Streicher, 2016)	✓	✓	✓
(Stavrev et al., 2018)		✓	✓
(Flores et al., 2019)		✓	✓
(Mizutani et al., 2021)			✓
(Olsson et al., 2015)			✓
(Carvalho, Bellotti, Hu, et al., 2015)			✓
(Scacchi, 2017)			✓
(Yessad et al., 2010)	✓	✓	
(Costa et al., 2016)		✓	✓
(Ismail & Belkhouche, 2019)		✓	✓

\*Equipo técnico: Diseñadores, desarrolladores o programadores

Con relación al criterio del **propósito o intención de la arquitectura** - Este criterio hace referencia al objetivo que caracteriza a la arquitectura, es decir, si la misma ha sido pensada para la formación de competencias, o para el desarrollo de juegos serios a nivel general. La [Tabla 7.33](#) presenta los resultados obtenidos.

Tal como se puede apreciar en la [Tabla 7.33](#), han sido 8 las propuestas (61.54%) que han sido pensadas con el propósito de ser utilizadas para el desarrollo de juegos serios a nivel general. Las otras 5 propuestas (38.46%) se han enfocado directamente a la creación de juegos enfocados a la formación de competencias.

*Tabla 7.33 Criterio - Propósito o intención de la arquitectura*

Metodología	Propósito de la arquitectura	
	Formación de competencias	Desarrollo
(Maggiorini et al., 2016)		✓
(El Mawas, 2014)	✓	
(Gutiérrez-Hernández et al., 2013)		✓
(Söbke & Streicher, 2016)	✓	
(Stavrev et al., 2018)	✓	✓
(Flores et al., 2019)	✓	✓
(Mizutani et al., 2021)		✓
(Olsson et al., 2015)		✓
(Carvalho, Bellotti, Hu, et al., 2015)		✓
(Scacchi, 2017)		✓
(Yessad et al., 2010)	✓	
(Costa et al., 2016)	✓	
(Ismail & Belkhouche, 2019)	✓	

La Tabla 7.34 presenta los resultados obtenidos con relación al criterio de **nivel de complejidad**, el cual permite determinar si la arquitectura propuesta es amigable con usuarios no expertos en el área de tecnologías.

*Tabla 7.34 Criterio - Nivel de complejidad*

Metodología	Nivel de complejidad de la arquitectura*		
	Bajo	Medio	Alto
(Maggiorini et al., 2016)			✓
(El Mawas, 2014)		✓	
(Gutiérrez-Hernández et al., 2013)		✓	
(Söbke & Streicher, 2016)		✓	
(Stavrev et al., 2018)		✓	
(Flores et al., 2019)		✓	
(Mizutani et al., 2021)	✓		
(Ollsson et al., 2015)		✓	
(Carvalho, Bellotti, Hu, et al., 2015)	✓		
(Scacchi, 2017)	✓		
(Yessad et al., 2010)	✓		
(Costa et al., 2016)		✓	
(Ismail & Belkhouche, 2019)		✓	

\***Nivel bajo** (conocimientos muy básicos en herramientas de ofimática o nulos), **nivel medio** (conocimientos avanzados en diferentes herramientas tecnológicas y, **nivel alto** (dominio de lenguajes de programación).

Según los datos de la Tabla 7.34, 8 de las propuestas (61.54%) presentan un nivel de complejidad medio. Luego, 4 de las propuestas registran un nivel bajo y, solo 1 arquitectura presenta un nivel alto de complejidad. Esto podría significar que, actualmente se comienza a pensar en el desarrollo de piezas de software que pueden ser utilizadas y adaptadas acorde a las necesidades de los usuarios, sin que los mismos requieran de conocimientos avanzados o dominio de lenguajes de programación para poder utilizarlos.

#### 4.17.4. Aspectos de análisis

La Tabla 7.35 muestra los resultados de la aplicación del criterio **tipo de producto por desarrollar**, el cual permite identificar si la propuesta se orienta al desarrollo de videojuegos comunes o juegos serios.

*Tabla 7.35 Criterio - Tipo de producto por desarrollar*

Metodología	Tipo de producto por desarrollar		
	Juego serio	Videojuego tradicional	Otro
(Maggiorini et al., 2016)	✓	✓	
(El Mawas, 2014)	✓		

Metodología	Tipo de producto por desarrollar		
(Gutiérrez-Hernández et al., 2013)	√		
(Söbke & Streicher, 2016)	√		
(Stavrev et al., 2018)			√
(Flores et al., 2019)	√		
(Mizutani et al., 2021)			√
(Ollsson et al., 2015)		√	
(Carvalho, Bellotti, Hu, et al., 2015)	√		
(Scacchi, 2017)	√		
(Yessad et al., 2010)	√		
(Costa et al., 2016)	√		
(Ismail & Belkhouche, 2019)			√

Según los datos de la Tabla 7.35, 10 de las propuestas analizadas (77% aproximadamente) fueron pensadas para el desarrollo de juegos serios. Esto quiere decir que, actualmente se están dando pasos en esta área. Con relación al criterio de **resultados de utilización obtenidos**, se obtuvo como resultado que 12 propuestas (92.30%) no registran resultados académicos y científicos que respalden los hallazgos obtenidos a partir de la implementación de la propuesta arquitectónica.

Finalmente, con relación a las **Guías o ayudas disponibles**, se identificó que 10 de las propuestas analizadas (77% aproximadamente) no ofrece algún tipo de ayuda que puede brindar la arquitectura a los usuarios (manuales, tutoriales, guías de apoyo, preguntas frecuentes, sugerencias, entre otros).

#### 4.18. Conclusiones del capítulo

De acuerdo con los resultados obtenidos durante el análisis de las propuestas arquitectónicas o investigaciones seleccionadas, se ha elaborado un **checklist** con las principales consideraciones a tomar en cuenta al diseñar una propuesta arquitectónica para el diseño y desarrollo de videojuegos, en particular, juegos serios:

- ✓ **Idioma** - Considerar que la propuesta esté implementada en un idioma universal, principalmente, en inglés o español.
- ✓ **Nivel educativo** - Considerar que la propuesta este pensada para un nivel educativo superior/universitario.

- ✓ **Fundamentación** - Considerar una arquitectura orientada a servicios.
- ✓ **Tipo aplicación a la que está dirigida la arquitectura** - Considerar que la ejecución de esta sea pensada para ejecutarse en al menos 2 tipos de aplicaciones, donde la aplicación web sea una de ellas.
- ✓ **Reutilización de componentes** - Considerar la reutilización, máxime si se orientara a una aplicación web.
- ✓ **Destinatarios de la arquitectura** - Considerar al equipo técnico (desarrolladores o programadores) como posibles destinatarios de la propuesta arquitectónica o bien, el usuario final, en este caso el profesorado.
- ✓ **Propósito o intención de la arquitectura** - Considerar que la propuesta arquitectónica permita el diseño y desarrollo de juegos serios orientados a la formación de competencias tecnológicas.
- ✓ **Nivel de complejidad** - Considerar para el desarrollo de la arquitectura un nivel medio (conocimientos avanzados en diferentes herramientas tecnológicas).
- ✓ **Identificar claramente el tipo de producto por desarrollar** - Considerar claramente que tipo de producto se va a desarrollar (JS o videojuego tradicional).
- ✓ **Ofrecer guías o ayudas** - Es necesario que los posibles usuarios cuenten con mínimo un manual de ayuda, video tutorial o bien, un foro de discusión.

Al igual que en las metodologías analizadas en el Capítulo VI, no se logra identificar de forma concreta en la revisión de literatura una propuesta arquitectónica para el desarrollo de juegos serios. Se concluye que el conjunto de arquitecturas o propuestas analizadas, servirán de guía en términos de las consideraciones que se

podrían tomar en cuenta al proponer este tipo de arquitecturas, donde la intensión pedagógica sea la formación de competencias tecnológicas.

## CAPÍTULO VIII. PROPUESTA METODOLÓGICA Y ARQUITECTÓNICA PARA EL DISEÑO DE JUEGOS SERIOS

## 8.1 Introducción

En este capítulo se describe a detalle la metodología INTEGRA+506 (INTEGRA por la integración de metodologías de diseño y arquitectónicas y +506 por ser el código del país de origen de la metodología) la cual ha sido propuesta por el tesista con el fin de integrar aspectos metodológicos y arquitectónicos en el diseño de juegos serios para potenciar en el profesorado la formación de competencias tecnológicas. La metodología INTEGRA+506 fue propuesta con el fin de guiar paso a paso un proceso planificado y detallado para el diseño de juegos serios orientados a formar al profesorado en competencias tecnológicas.

## 8.2 Pasos metodológicos y arquitectónicos para el diseño de juegos serios

En este apartado se describen los pasos metodológicos y arquitectónicos recomendados para el diseño de juegos orientados a la formación de competencias tecnológicas en el profesorado.

Para el diseño de la propuesta metodológica y arquitectónica denominada INTEGRA+506, en primer lugar, se tomó como base o referencia la propuesta diseñada por el tesista en la maestría y especializaciones (Sandí-Delgado, 2019; Sandí-Delgado et al., 2018, 2022).

En segundo lugar, se consideraron las principales recomendaciones existentes en la literatura, las cuales fueron obtenidas a través de la revisión bibliográfica, donde se consideró las recomendaciones de diversos autores (All et al., 2016; Almerich et al., 2016; Archuby et al., 2018; Boyle et al., 2011, 2012, 2016; Carvalho, Bellotti, Berta, De Gloria, Islas, et al., 2015; Connolly et al., 2012; El Mawas, 2014; Evans & Massa, 2017; Flores et al., 2019; Gutiérrez-Hernández et al., 2013; Hainey et al., 2016; Ismail, 2018; Ismail & Belkhouche, 2019; Maggiorini et al., 2016; Marfisi-Schottman, 2012; Marfisi-Schottman et al., 2010; Massa et al., 2017; Ministerio de Educación de Chile, 2006; Mizutani et al., 2021; Ollsson et al., 2015; Padilla-Zea et al., 2015; Prendes et al., 2010, 2018; Prendes & Gutiérrez, 2013; Sandí-Delgado, 2019, 2020; Sandí-Delgado & Bazán, 2017, 2019, 2021; Sandí-Delgado & Sanz, 2018a, 2018b; Scacchi, 2017; Silva, 2012; Söbke & Streicher, 2016; Spinelli et al., 2016, 2017; Stavrev et al., 2018; Yessad et al., 2010).

En tercer lugar, a nivel metodológico INTEGRA+506 se fundamenta a partir del estudio, identificación y análisis de metodologías y modelos propuestos para el diseño e implementación de juegos serios y video juegos que han sido comúnmente más utilizados, tales como EMERGO (Nadolski et al., 2008), EDoS (Tran et al., 2010), SAVIE (Sauvé, 2009), DODDEL (McMahon, 2009), VGSC (Padilla-Zea, 2011), MECONESIS (Cano, 2016), MPIU+a (Granollers, 2004) y MPDSG (Evans et al., 2016).

En cuarto lugar, a nivel de arquitectura INTEGRA+506 se fundamentan a partir de la revisión y análisis de la literatura existente en temas relacionados con el objetivo de investigación (Maggiorini et al., 2016; El Mawas, 2014; Gutiérrez-Hernández et al., 2013; Söbke & Streicher, 2016; Stavrev et al., 2018; Flores et al., 2019; Mizutani et al., 2021; Ollsson et al., 2015; Carvalho, Bellotti, Hu, et al., 2015; Scacchi, 2017; Yessad et al., 2010; Costa et al., 2016; Ismail & Belkhouche, 2019).

Dicha metodología sienta las bases para la integración de metodologías y arquitecturas enfocadas en el diseño de juegos serios. Cada paso metodológico propuesto se sustenta con sus respectivas citas o referencias.

### **8.3 Propuesta metodológica y arquitectónica INTEGRA+506**

En el presente apartado se aborda en detalle los pasos metodológicos y arquitectónicos presentes en la metodología INTEGRA+506 la cual guía paso a paso el diseño de juegos serios orientados a la formación de competencias tecnológicas, siguiendo las principales consideraciones teóricas en relación con el desarrollo de software, metodologías y arquitecturas para el diseño y desarrollo de juegos serios. Además, como un plus, se ha considerado la inclusión de algunas de las características funcionales que los juegos serios educativos adquirirían al ser utilizados como software de servicio, mismas descritas en detalle en la investigación previa realizada por el tesis (Sandí-Delgado & Bazán, 2017, 2019).

Por tanto, se propone una metodología compuesta por 7 fases, donde cada fase incluye diferentes pasos metodológicos y arquitectónicos (PM&A) que guía el diseño del prototipo no funcional del juego por diseñar. Para la definición de las fases de la metodología se consideraron los aspectos teóricos del desarrollo de software (capítulo V), los aspectos analizados en las metodologías de juegos serios

(capítulo VI) y los aspectos identificados en la revisión de las arquitecturas (capítulo VII).

No está demás indicar que, tanto las fases como los pasos metodológicos podrían ser considerados de forma distinta dependiendo de la experiencia o área de formación de la persona o investigador que las analiza.

### **8.3.1 Fase 1. Consideraciones generales**

Esta primera fase es una de las más importantes de la metodología, ya que se requiere definir los roles o actores participantes durante el desarrollo del prototipo, así como establecer las competencias a potenciar y el tipo de aplicación que se utilizará para ejecutarlo.

#### **FM&A01 Definir roles o actores participantes**

El propósito de este paso metodológico consiste en identificar quienes pueden ser el equipo interdisciplinario, es decir, las personas expertas más recomendadas para participar en la definición de la propuesta del juego serio, tales como expertos pedagógicos, en el tema, de contenido, desarrolladores, pedagogos, entre otros (Spinelli & Massa, 2018a), máxime que la definición de los requerimientos de usuario son influyentes en relación con la calidad de la pieza de software por desarrollar (Alshamrani & Bahattab, 2015; Alshazly et al., 2014).

#### **FM&A02 Seleccionar competencias – Definir objetivos**

Acorde con las investigaciones realizadas por el tesista (Sandí-Delgado, 2019; Sandí-Delgado et al., 2018, 2022; Sandí-Delgado & Sanz, 2020), al diseñar un juego serio para la formación de competencias tecnológicas, se debe considerar en primera instancia la definición de los requerimientos, luego, se debe definir claramente los objetivos del juego, es decir, seleccionar previamente las competencias que se busca potenciar. En este sentido, existen estudios previos que han definido y clasificado las competencias en dimensiones e indicadores, tal es el caso del Ministerio de Educación de Chile (2006), las investigaciones realizadas por Prendes & Gutiérrez (2013) en España, por el Ministerio de Educación Nacional Colombiano (Campo et al., 2013; Hernández et al., 2016), estudios realizados por

(Silva, Mirand-, Gisbert, Morales, & Onetto, 2016), en Uruguay, Paraguay (Arevalos, 2014) y por el Consejo Universitario de la Universidad de Costa Rica (UCR) (UCR, 2004), entre otros.

Lo anterior, coincide con lo indicado por (Angeli & Valanides, 2005, 2009; Archambault & Barnett, 2010; Blackwell, Lauricella, & Wartella, 2016; Graham, 2011; Scherer, Tondeur, & Siddiq, 2017; Yeh, Hsu, Wu, & Chien, 2017; Scherer, Tondeur, Siddiq, & Baran, 2018) los cuales agregan que para integrar las TIC en los procesos formativos es necesario poseer habilidades *tecnológicas, pedagógicas y disciplinares*, competencias que han sido también propuestas por el modelo de Conocimiento Técnico Pedagógico del Contenido (TPACK, *Technological Pedagogical Content Knowledge*). El modelo TPACK según (Almerich et al., 2016) es un modelo de extensión del conocimiento del contenido pedagógico, al cual se le ha adicionado el contenido tecnológico (Sandí-Delgado et al., 2022).

Este paso inicial es sumamente importante debido a que se define el objetivo o propósito a alcanzar.

### **FM&A03      Seleccionar el tipo de aplicación**

Posterior a la selección de las competencias tecnológicas a potenciar en el profesorado a través del prototipo del juego serio por diseñar, se debe realizar la selección del tipo de aplicación sobre la cual se desarrollará el juego serio (Hainey et al., 2016), para ello, se debe considerar aquellos que hayan sido utilizados con mayor frecuencia para la ejecución de juegos serios. En este sentido, en investigaciones realizadas por (Boyle et al., 2011; Giannakos, 2013; Hainey et al., 2016; Sandí-Delgado et al., 2022; Sandí-Delgado & Bazán, 2017; Sandí-Delgado & Sanz, 2018a), se consideran 5, entre ellas: para PC, las orientadas a trabajar en línea (web), video consolas, móviles y finalmente, las orientadas a la virtualidad.

El tipo de aplicación a seleccionar va a depender de los objetivos propuestos y de los posibles recursos con que cuente la institución y el profesorado directamente involucrado. El prototipo de juego serio requiere de poder acceder a la aplicación desde cualquier computadora o dispositivo conectado a internet, el cual podría utilizar diferentes navegadores web (Albert & Torres, 2022a).

#### **8.3.2 Fase 2. Lenguaje de programación**

Para el diseño de una pieza de software y, en particular, el prototipo de un juego serio, se requiere de la definición de un paradigma y tipo de lenguaje de programación (IEEE Computer Society, 1990; Prasad & Verma, 2016).

#### **FM&A04      Paradigma de programación**

Para el prototipo no funcional del juego serio por desarrollar, se requiere que se defina un paradigma de programación, ya que este indicará el método para su construcción, además, permitirá organizar las tareas que debe llevar a cabo la pieza de software, además, los mismos definen los estilos de programación (INET, 2016; Vaca, 2012). Se podría elegir entre paradigmas de programación no estructurada, programación estructurada y programación orientada a objetos (mismos descritos en detalle en el capítulo 5).

#### **FM&A05      Tipo de lenguaje de programación**

En relación con el lenguaje de programación para crear la aplicación del prototipo no funcional, se puede utilizar una variedad de lenguajes enfocados en el desarrollo web. La estructura de una aplicación web consiste en el *Frontend* (la interfaz orientada al usuario), y el *Backend* (código interno para el manejo de datos del Frontend) acorde con lo indicado por (Gagliardi, 2021; Jones, 2020; Slivnik, 2022).

- A. Lenguajes para el desarrollo **Frontend**: Comúnmente se utiliza para el desarrollo de Frontend lenguajes como JavaScript (brinda dinamismo a los sitios web), CSS (aporta estilos, diseños y formatos a la página web) y el Lenguaje de Marcado de Hipertexto (HTML, *HyperText Markup Language*) el cual provee una estructura de etiquetas y maquetado para el desarrollo del sitio web.
- B. Lenguajes para el desarrollo del **Backend**: Existe variedad de lenguajes para desarrollo del *Backend*, entre ellos se encuentra PHP (*Hypertext Preprocessor*), Python, Java, C++, JavaScript, entre otros, los cuales pueden ser utilizados para el desarrollo del prototipo.

Dependerá del desarrollador cual lenguaje a utilizar para el desarrollo del prototipo, máxime que la selección de este queda sujeta a la experiencia del usuario (desarrolladores) y de los objetivos por alcanzar.

### 8.3.3 Fase 3. Procesos de software

Acorde con lo indicado por (Sommerville, 2016), se debe definir un proceso de software para guiar todas las actividades relacionadas con la producción de la pieza de software, en este caso, el modelo de desarrollo y el diseño centrado en el usuario.

#### FM&A06     Modelo de desarrollo de software

Para el diseño de la pieza de software por desarrollar, se podría utilizar un modelo de cascada, modelo de proceso incremental, modelo de prototipos, modelo espiral, modelo de desarrollo de aplicación rápida o bien, un modelo de ciclo de vida ágil.

#### FM&A07     Diseño centrado en el usuario

Para el diseño del prototipo se requiere de un diseño centrado en el usuario para ofrecer una mayor satisfacción y experiencia a nivel del usuario (UX). Lo anterior, debido a que se requiere que los usuarios influyan sobre las decisiones de diseño y sobre los objetivos pretendidos por alcanzar con el prototipo.

En relación con los estilos de componentes a nivel de interfaz para prototipos, se recomienda que para el manejo del diseño y presentación de la aplicación, se definan los estilos que presentará el sitio web, dentro del diseño el intérprete que asignará los estilos es el lenguaje Hojas de Estilo en Cascada (CSS<sup>9</sup> *Cascading Style Sheets*), para la utilización de este, existe variedad de opciones a elegir, entre ellos *Frameworks* sencillos de utilizar, tales como Bootstrap<sup>10</sup>, Bulma<sup>11</sup>, MDBBootstrap<sup>12</sup>, que permiten brindar una identidad distinta al estilo, según el *Frameworks* elegido. De igual manera se podría utilizar CSS en su versión pura

<sup>9</sup> <https://www.w3schools.com/css/>

<sup>10</sup> <https://getbootstrap.com/>

<sup>11</sup> <https://bulma.io/>

<sup>12</sup> <https://mdbbootstrap.com/>

“vanilla”, la cual se caracteriza por ser ligera, adaptable y de código abierto (Melgarejo-Torralba et al., 2022).

#### **8.3.4 Fase 4. Ingeniería de requerimientos**

Esta fase es de suma importancia en el desarrollo de piezas de software (Sommerville, 2016), las partes involucradas o interesadas deben definir los requerimientos generales, ya que a como una buena especificación de requerimientos es determinante en la calidad del software (Aurum & Wohlin, 2005; Gómez-Fuentes, 2011). Para ello, se debe considerar los requerimientos funcionales y no funcionales, así como definir claramente los casos de uso, las reglas de negocio y la ficha del juego.

##### **FM&A08 Requerimientos funcionales y no funcionales**

Para el diseño de software se requiere definir claramente los servicios que el sistema debe proporcionar e indicar cuales serían las restricciones de esta.

##### **FM&A09 Casos de uso**

Para el diseño del prototipo no funcional del juego serio, se requiere la descripción de las acciones o actividades que permitan concretar el proceso del diseño final (Sommerville, 2016).

##### **FM&A10 Reglas de negocio**

Para el diseño del prototipo no funcional del juego serio, se requiere la definición de las reglas de negocios, las cuales tienen como objetivo definir restricciones o límites que permitan alcanzar el objetivo del proyecto.

##### **FM&A11 Ficha de juego**

Los juegos serios usualmente, suelen tener asociada una ficha de juego, en la cual se cuente brevemente su **historia** (de qué trata el juego), así como ofrecer las **indicaciones del cómo jugarlo** y, finalmente, incluir el aspecto más caracterizante de los juegos serios, el cual corresponde a la **estrategia de ludificación**, es decir, el juego describir la metodología de ludificación que utiliza (medallas, ranking,

insignia, puntajes, premiación, entre otros) (Sandí-Delgado & Sanz, 2020), lo cual es un aspecto relevante para considerar, debido a que los juegos serios se caracterizan por la intención pedagógica y, por incorporar estrategias lúdicas (Sandí-Delgado et al., 2018; Sandí-Delgado & Bazán, 2019; Sandí-Delgado & Sanz, 2018b, 2020).

### 8.3.5 Fase 5. Diseño de software – Aspectos arquitectónicos

En el diseño de software se requiere generar una estructura que permita guiar el diseño de la misma, es decir, debe permitir describir el “como” se logra cada función propuesta del software mediante una especificación de propiedades de objeto (Ralph & Wand, 2007). Entre las especificaciones, se requiere describir como mínimo el estilo arquitectónico (con todos sus componentes) y el diagrama de flujo de datos.

#### FM&A12 Estilo arquitectónico

El estilo arquitectónico puede estar integrado por diferentes componentes, los cuales varían acorde con la experiencia del desarrollador, el objetivo del juego y tipo de aplicación a la que está orientada la pieza de software.

- ✓ **Modelo, Vista, Controlador** – En relación con el modelo, vista, controlador para prototipos, se requiere para separar de manera lógica la estructura de programación en una segregación por capas (Fragoso-Díaz et al., 2008; López Caballero et al., 2019).
- ✓ **Microservicios (web services)** – Se requiere la implementación de un web services para el consumo de recursos almacenados en la base de datos donde estará alojada la información del juego, tales como información general del usuario y métricas: tiempo y puntuaciones (Fragoso-Díaz et al., 2008; López Caballero et al., 2019).
- ✓ **Entorno de desarrollo** - En relación con los estilos de componentes a nivel de interfaz para prototipos, se requiere seleccionar un Entorno de Desarrollo

Integrado (IDE, *Integrated Development Environment*) adecuado al lenguaje que se va a utilizar (Bazán et al., 2017; Yessad et al., 2010; Yulianto et al., 2017), en este sentido, las opciones a utilizar podrían ser amplias y con alta variedad de software entre el cual elegir. Dentro de las opciones más utilizadas en el mercado para el desarrollo web están: Visual Studio Code<sup>13</sup> debido a su alta flexibilidad para trabajar como lenguaje, WebStorm<sup>14</sup> y SublimeText<sup>15</sup>, entre otros (Janssen et al., 2019; Yun et al., 2020).

- ✓ **Framework o Librerías** - Para el desarrollo del prototipo no funcional del juego serio se podría utilizar un *Frameworks*, el cual permite el desarrollo a través de un estructura o esqueleto (es decir, es un marco de trabajo), el cual aporta flexibilidad para el manejo tanto del *Backend* como del *Frontend*, debido a que estos marcos de trabajo cuentan con librerías creadas por la misma comunidad de desarrolladores que los utilizan (Leung & Cockburn, 2021), estos podrían facilitar el desarrollo de piezas de software dependiendo de las circunstancias a implementar (Cabrera, 2017). En la actualidad, existe una variedad de Frameworks para JavaScript tales como React<sup>16</sup>, AngularJS<sup>17</sup>, Vue.js<sup>18</sup> (Albert & Torres, 2022b), cada uno posee sus características particulares a nivel de usabilidad y robustez, por ejemplo; Angular y React han sido utilizados por Uber, Google, Paypal y The New York Times.
  
- ✓ **Gestor de base de datos** - Para el desarrollo del prototipo no funcional del juego serio se requiere de la utilización de un Sistema Gestor de Base de Datos (DBMS, *Database Management System*) ya que se requiere administrar el acceso por parte de los usuarios a la base de datos, así como permitir la realización de consultas y generación de informes en relación con los avances de los jugadores (Hueso-Ibáñez, 2015). El DBMS usualmente cumple con el objetivo de servir de interfaz entre esta, el usuario y las aplicaciones (Díaz-Erazo et al., 2022). Para el almacenamiento de los datos se podría utilizar varios

<sup>13</sup> <https://code.visualstudio.com/>

<sup>14</sup> <https://www.jetbrains.com/webstorm/>

<sup>15</sup> <https://www.sublimetext.com/>

<sup>16</sup> <https://es.reactjs.org/>

<sup>17</sup> <https://angular.io/>

<sup>18</sup> <https://vuejs.org/>

gestores, tales como MySQL<sup>19</sup>, SQL<sup>20</sup>, Oracle<sup>21</sup>, entre otras. Estos gestores permiten almacenar ordenadamente la información generada por los jugadores.

- ✓ **Alojamiento** - En el prototipo no funcional del juego serio se requiere alojar la aplicación en la web para garantizar su disponibilidad y alcance al usuario. Lo más utilizado actualmente, son la contratación de servicios de proveedores de computación en la nube (Cobeña, 2021). Algunas opciones para alojar la aplicación (juego, API, entre otros) son Azure<sup>22</sup>, AWS<sup>23</sup>, *Google Cloud Platform*<sup>24</sup>, entre otras (Albert & Torres, 2022a), las cuales usualmente ofrecen un costo económico por el servicio consumido de alojamiento. Igualmente, se podría valorar la opción de buscar alternativas gratuitas como GitHub<sup>25</sup>.
- ✓ **Autentificación de usuarios** - En el prototipo no funcional del juego serio se requiere de la autentificación de usuarios, la cual se podría realizar de distintas maneras; utilizar librerías como (JWT<sup>26</sup>, *Json Web Token*) o Auth0<sup>27</sup>, las cuales brindan seguridad garantizada en el control de acceso de los usuarios (Díaz-Erazo et al., 2022). Este proceso es requerido para garantizar la privacidad, integridad y calidad de los datos de cada jugador, así como mantener un registro individual de la actividad y alcances de cada usuario.
- ✓ **Versionado** - El prototipo de juego serio requiere ofrecer una aplicación simple, que se mantenga una única versión compartida para todos los clientes (Sandí-Delgado & Bazán, 2017). Se podrían utilizar herramientas existentes para el control de versiones, tales como Git (sistema de control de versiones distribuido), el cual ofrece diferentes clientes tales como GitHub, GitLab, entre otros, Subversión (sistema de control de versiones centralizado) y Mercurial (sistema de control de versiones multiplataforma), siendo este último el menos

<sup>19</sup> <https://www.mysql.com/>

<sup>20</sup> <https://www.microsoft.com/es-es/sql-server/sql-server-downloads>

<sup>21</sup> <https://www.oracle.com/es/>

<sup>22</sup> <https://azure.microsoft.com/es-mx/>

<sup>23</sup> [https://aws.amazon.com/es/?nc2=h\\_lq](https://aws.amazon.com/es/?nc2=h_lq)

<sup>24</sup> <https://cloud.google.com/>

<sup>25</sup> <https://github.com/>

<sup>26</sup> <https://jwt.io/>

<sup>27</sup> <https://auth0.com/es>

utilizado (Albert & Torres, 2022a). La diferencia más importante entre Git y Subversión consiste en que, en los sistemas centralizados (Subversión) se mantiene únicamente una copia de la aplicación y las copias permanecen respaldadas en un solo lugar. En los sistemas distribuidos que hacen uso de Git, se mantiene una copia del proyecto original en cada equipo local y los cambios se actualizan o suben al repositorio en línea, los cuales posteriormente deben ser descargados de dicho repositorio para actualizar la versión de cada uno de los desarrolladores (Albert & Torres, 2022a). El versionado de los diferentes repositorios (frontend, backend) de las aplicaciones web se pueden actualizar sin interrumpir su flujo, al desplegar la aplicación con la nueva versión del repositorio la misma se actualiza con sus nuevas funcionalidades o correcciones.

#### **FM&A13      Diagramas de clases (UML) y Diagrama entidad relación (ER)**

En el diseño del prototipo se requiere de la definición de un diagrama de clases y un diagrama entidad relación debido a que este permite identificar claramente cual será el flujo de la información durante el desarrollo de la pieza de software. Para ello, se pueden utilizar herramientas gratuitas tales como Draw.io<sup>28</sup>, StarUML.io<sup>29</sup>, Diagrams.net<sup>30</sup>, o bien de paga como Visio<sup>31</sup>. En estos diagramas se hace uso de simbología tal como el círculo, diferentes tipos de flechas, rectángulos, texto entre otros, los cuales permiten visualizar la entrada y salida de datos (Sommerville, 2016).

#### **8.3.6 Fase 6. Codificación del software**

En el diseño del prototipo se sugiere contemplar la codificación del software, debido a que permite traducir todas las especificaciones diseñadas en código fuente, haciendo más fácil el leer y comprender la pieza de software desarrollada (Prasad & Verma, 2016). Lo anterior, permite obtener un estilo de programación,

<sup>28</sup> <https://app.diagrams.net/>

<sup>29</sup> <https://staruml.io/>

<sup>30</sup> <https://www.diagrams.net/>

<sup>31</sup> <https://www.microsoft.com/es-ww/microsoft-365/visio/flowchart-software>

pautas de codificación, técnicas de codificación estructuradas y técnicas de verificación del código. En esta fase se analiza tanto las características del código de software como la reutilización de sus componentes (Sommerville, 2016).

#### **FM&A14      Características del código de software**

Para el diseño del prototipo se sugiere que el código del software esté escrito en congruencia con los requisitos del usuario, es decir, apegado a los estándares y buenas prácticas de programación, eficiente en sus algoritmos, con manejo de excepciones, validaciones y medidas de seguridad, garantizando un mínimo margen de errores (Prasad & Verma, 2016). Se pueden implementar algunas de las siguientes medidas: simplicidad de código (simple y conciso), modularidad (dividido en varios módulos), diseño (estilo específico y consistente), eficiencia (uso óptimo de los recursos disponibles), claridad (simplicidad, legibilidad y modularidad), accesibilidad (fácilmente disponibles y accesibles) o estabilidad (funcionar correctamente en diferentes plataformas sin afectar su diseño y consistencia).

#### **FM&A15      Reutilización de componentes**

En el diseño del prototipo se sugiere contemplar la reutilización de código y componentes dinámicos de software, máxime si se llega a pensar en una aplicación web, debido a que facilita el mantenimiento, escalabilidad y funcionalidad de la aplicación (Prasad & Verma, 2016; Ralph & Wand, 2007). La reutilización del software es posible de implementar en diferentes niveles: Nivel de abstracción, nivel del objeto, nivel del componente y nivel del sistema (Sommerville, 2016).

#### **8.3.7 Fase 7. Validación de la metodología**

En el desarrollo de software se recomienda que las propuestas metodológicas sean validadas por profesionales y expertos en el área, por ello, en este caso, se recomienda hacer uso del juicio de expertos.

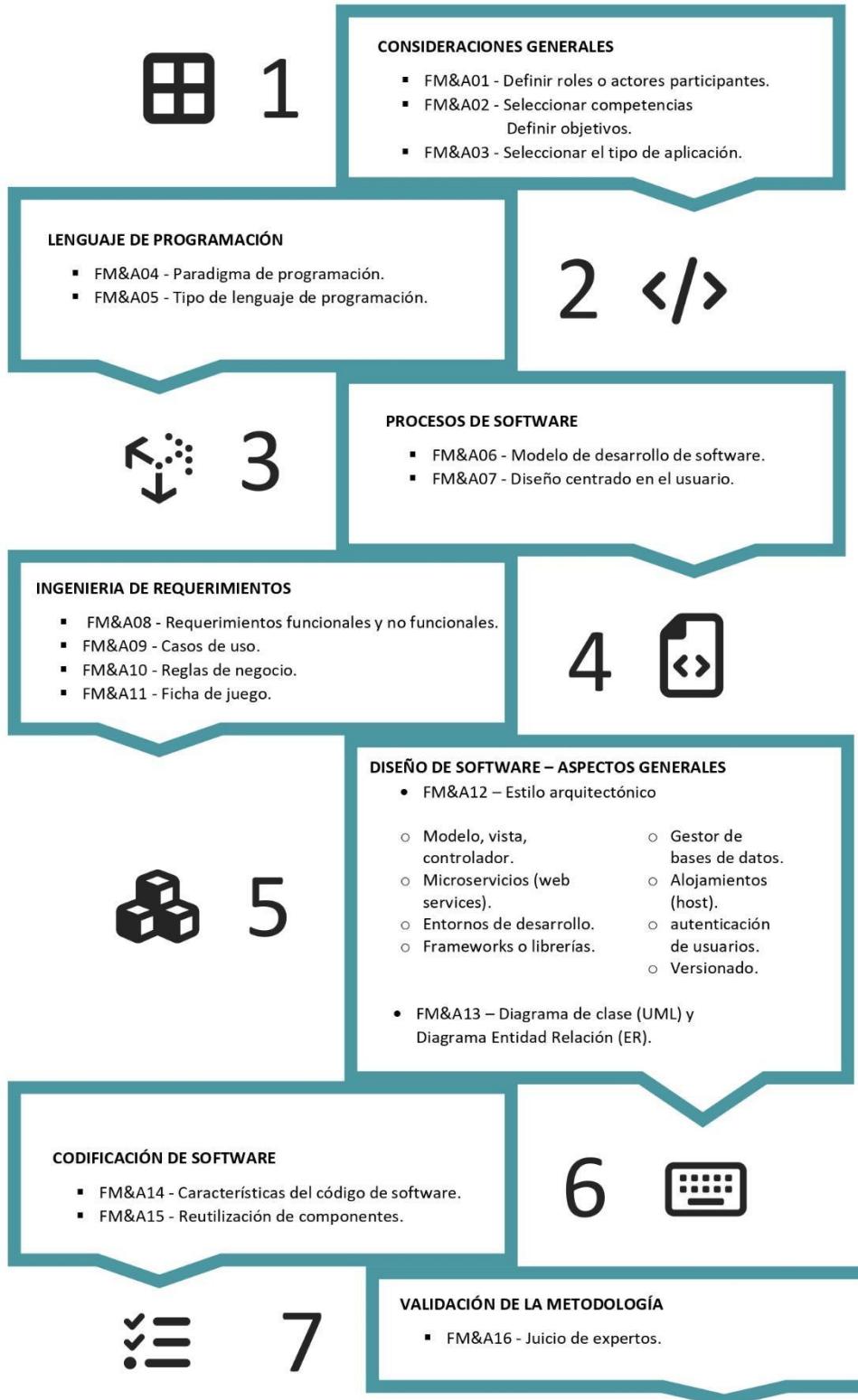
#### **FM&A16      Juicio de Expertos**

Para la validación de los pasos propuestos en la metodología INTEGRA+506 se recomienda utilizar el juicio de expertos, quienes revisan, analizan y validan cada

uno de los pasos, tecnologías y herramientas propuestas. Se recomienda especialista de áreas del conocimiento relacionadas con el campo de la informática, computación, diseño gráfico, pedagogía, entre otros, quienes tendrían trayectoria reconocida en sus campos y podrían ofrecer juicios y valoraciones al respecto.

#### **8.4 Conclusiones del capítulo**

En este capítulo se describió en detalle la metodología INTEGRA+506 propuesta para integrar aspectos metodológicos y arquitectónicos en el diseño de juegos serios para potenciar en el profesorado la formación de competencias tecnológicas. La metodología está compuesta por 7 fases y 15 pasos metodológicos y arquitectónicos. En la Figura 8.1, se muestra a manera de síntesis las secuencias de dichos pasos descritos.



**Figura 8. 1 Secuencia metodológica para el diseño de juegos serios**

# CAPÍTULO IX. PROTOTIPO NO FUNCIONAL DE SKILLNET

## 9.1. Introducción

En el presente capítulo se aborda y describe en detalle el proceso de implementación del prototipo no funcional del juego serio propuesto. Es decir, se aborda y describe en detalle la metodología **INTEGRA+506**, la cual permite guiar paso a paso la implementación del juego serio denominado **SkillNet** (prototipo no funcional).

En la próxima sección se aborda y describe la implementación de la metodología **INTEGRA+506**, para obtener como producto final el prototipo del juego serio **SkillNet**.

## 9.2. Implementación de INTEGRA+506 + SkillNet

La Tabla 9. 1 resume las fases metodológicas y arquitectónicas de **INTEGRA+506**.

**Tabla 9. 1** Fases y pasos metodológicos y arquitectónicos de INTEGRA+506

Fase	Pasos metodológicos y arquitectónicos
Fase 1. Consideraciones generales	FM&A01 Definir roles o actores participantes. FM&A02 Seleccionar competencias – Definir objetivos. FM&A03 Seleccionar el tipo de aplicación.
Fase 2. Lenguaje de programación	FM&A04 Paradigma de programación. FM&A05 Tipo de lenguaje de programación.
Fase 3. Procesos de software	FM&A06 Modelo de desarrollo de software. FM&A07 Diseño centrado en el usuario.
Fase 4. Ingeniería de requerimientos	FM&A08 Requerimientos funcionales y no funcionales. FM&A09 Casos de uso. FM&A10 Reglas de negocio. FM&A11 Ficha de juego.
Fase 5. Diseño de software – aspectos arquitectónicos	Estilo arquitectónico: ✓ Modelo, vista, controlador. ✓ Microservicios (web service). ✓ Entornos de desarrollo. ✓ Framework o Librerías. ✓ Gestor de base de datos.

Fase	Pasos metodológicos y arquitectónicos
	<ul style="list-style-type: none"> <li>✓ Alojamiento.</li> <li>✓ Autentificación de usuarios.</li> <li>✓ Versionado.</li> </ul>
	FM&A13 Diagramas de clases (UML) y Diagrama Entidad Relación (ER).
Fase 6. Codificación del software	FM&A14 Características del código de software. FM&A15 Reutilización de componentes
Fase 7. Validación de la metodología	FM&A16 Juicio de expertos

Seguido, se aborda en detalle cada uno de los pasos desarrollados en la implementación de la propuesta para obtener el diseño del prototipo no funcional del juego serio denominado SkillNet.

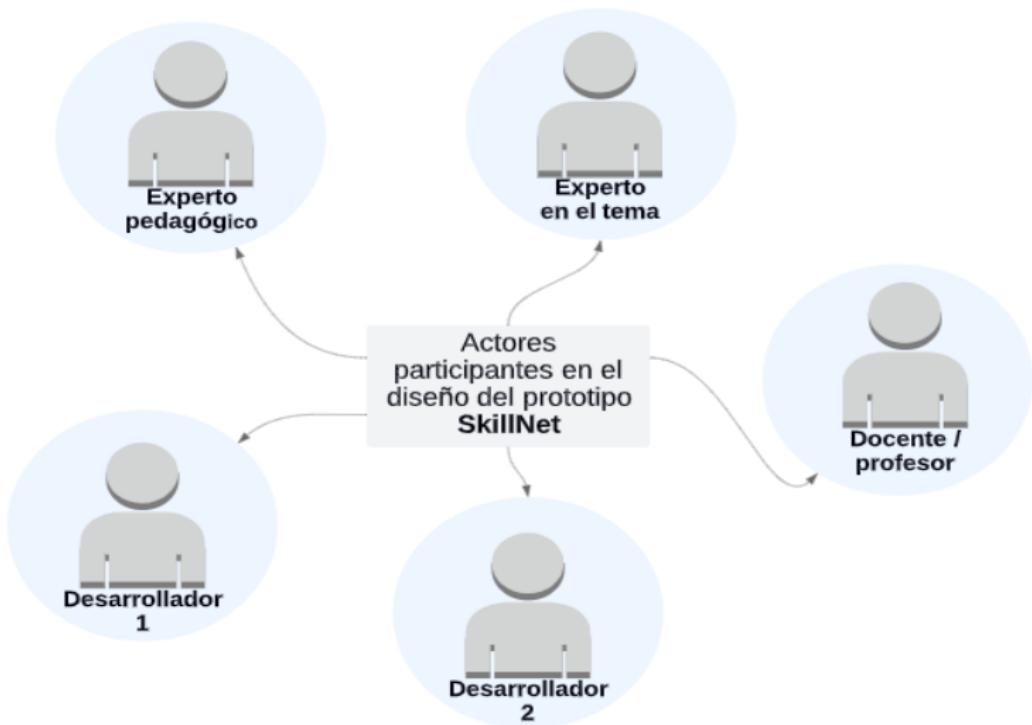
### 9.2.1. Fase 1. Consideraciones generales

Para el prototipo no funcional SkillNet, se consideraron los siguientes roles, competencias y tipo de aplicación:

#### FM&A01 Definir roles o actores participantes

Para esta fase inicial se procede con la identificación de los actores que intervendrán durante el desarrollo del prototipo no funcional de **SkillNet** acorde con lo indicado en las diferentes investigaciones desarrolladas por (Alshamrani & Bahattab, 2015; Prasad & Verma, 2016; Sommerville, 2016; Spinelli & Massa, 2018a).

La Figura 9. 1 muestra a los actores participantes en el diseño del prototipo, entre ellos: experto en el tema (conoce el problema a resolver), experto pedagógico (para garantizar la correcta definición de los objetivos), desarrollador 1, desarrollador 2 y docente (usuario final).



*Figura 9. 1 Actores participantes en el diseño del prototipo de SkillNet*

## FM&A02      **Seleccionar competencias – Definir objetivos**

Para el desarrollo del prototipo no funcional de **SkillNet**, se inició con la selección de las competencias tecnológicas que se requerían o buscaban potenciar en el profesorado a través del juego. En este sentido, se utilizó como referencia las investigaciones previas realizadas por el tesista (Sandí-Delgado et al., 2018, 2022; Sandí-Delgado & Bazán, 2019, 2021; Sandí-Delgado & Sanz, 2018b) y las propuestas por (Prendes, 2010; Prendes & Gutiérrez, 2013) en España. Así mismo, estas competencias han sido clasificadas en dimensiones e indicadores acorde con investigaciones realizadas por el (Ministerio de Educación de Chile, 2006).

La Tabla 9. 2 permite observar la clasificación de dichas competencias, las cuales han sido utilizadas como referencia para la elaboración del prototipo. En esta misma Tabla, se puede inferir que se hace uso de 3 dimensiones (investigación, gestión y, Docencia, investigación y gestión) con diferentes indicadores cada una.

**Tabla 9. 2 Competencias del profesorado en TIC en España**

<b>Dimensiones</b>	<b>Indicadores</b>
Investigación	Conoce y aplica los principios legales y éticos asociados al uso de información digital y TIC.
	Utiliza y promueve el uso de formatos abiertos para la publicación de contenidos digitales.
Gestión	Utiliza los recursos TIC que le proporciona la institución para llevar a cabo procesos de investigación.
Docencia, investigación y gestión	Conoce conceptos y componentes básicos asociados a las TIC.
	Maneja la información necesaria para la selección y utilización de recursos TIC.
	Aplica medidas de seguridad y prevención de riesgos en la operación de equipos tecnológicos y la salud de las personas.
	Es capaz de resolver las incidencias técnicas y sabe hacerles frente.
	Es capaz de aprender de forma autónoma el uso de herramientas y aplicaciones.
	Conoce el papel de las TIC en la formación de los titulados a los que da clase.
	Conoce las relaciones entre el currículo de su área de conocimiento y la forma de integrar las TIC en su práctica docente.
	Conoce diferentes estrategias metodológicas para integrar las TIC en su docencia.
	Conoce buenas experiencias educativas de su área de especialidad en la universidad que hagan uso de recursos TIC.
	Conoce buenas experiencias educativas en la universidad que, en general, hagan uso de recursos TIC.
	Conoce las implicaciones que la política educativa tiene en sus prácticas docentes en el aula, especialmente en lo relacionado con las TIC.
	Selecciona y utiliza herramientas y recursos TIC adecuados para el aprendizaje de los estudiantes.
	Selecciona y utiliza estrategias de enseñanza que implican el uso de TIC.
	Utiliza herramientas TIC para la producción de material didáctico.
	Utiliza las TIC para difundir su material didáctico.
	Emplea criterios de carácter pedagógico para seleccionar recursos TIC.
	Diseña actividades en las que se incorporan recursos TIC.
	Utiliza los servicios de apoyo a la implementación de TIC para la docencia proporcionados por la universidad.
	Utiliza diversas estrategias metodológicas con TIC.

Dimensiones	Indicadores
	Resuelve necesidades de aprendizaje con el uso de recursos TIC.
	Implementa actividades formativas en las que se incorporan recursos TIC.
	Utiliza sus habilidades comunicativas para favorecer la participación en entornos TIC.
	Utiliza las TIC en procesos de tutoría y en la evaluación de los aprendizajes.
	Utiliza TIC para evaluar procesos cognitivos complejos.
	Evalúa el efecto de sus prácticas docentes con TIC para incorporar las conclusiones en futuras experiencias.
	Actualiza permanentemente sus conocimientos respecto del desarrollo de las TIC y sus nuevas aplicaciones.
	Participa en proyectos de investigación e innovación educativa con TIC.
	Coordina o promueve en su unidad académica actividades apoyadas en el uso de las TIC.
	Participa en actividades de formación relacionados con las TIC.
	Participa en redes profesionales, que utilizan los recursos TIC para la docencia.
	Pertenece o promueve grupos de innovación e investigación en el uso de TIC para la docencia.
	Crea y mantiene un listado de sitios relevantes a su quehacer docente y desarrollo profesional.
	Utiliza fuentes diversas de información para su actualización en TIC y formación.

**Nota:** Tabla tomada de la información publicada por (Sandí-Delgado et al., 2018, pp. 70–78, 2022; Sandí-Delgado & Sanz, 2018a, pp. 27–28).

Para el diseño del prototipo no funcional de SkillNet, se han elegido 5 indicadores de la dimensión “docencia” de la Tabla 9. 1, mismos que se pueden observar en la Tabla 9. 3.

**Tabla 9. 3 Competencias e indicadores a potenciar en el profesorado**

Competencias seleccionadas a potenciar	
Dimensión	Indicadores
	Conoce conceptos y componentes básicos asociados a las TIC.
Docencia	Es capaz de aprender de forma autónoma el uso de herramientas y aplicaciones.

<b>Competencias seleccionadas a potenciar</b>	
<b>Dimensión</b>	<b>Indicadores</b>
	Utiliza diversas estrategias metodológicas con TIC.
	Implementa actividades formativas en las que se incorporan recursos TIC.
	Participa en actividades de formación relacionadas con las TIC.

**Nota:** Tabla tomada de la información publicada por (Sandí-Delgado et al., 2018, pp. 70–78, 2022; Sandí-Delgado & Sanz, 2018a, pp. 27–28)

Acorde con lo señalado por (Sandí-Delgado et al., 2018, 2022; Sandí-Delgado & Sanz, 2018a), el personal docente para lograr usar e integrar correctamente las TIC en la docencia, debe ser tecnológica y pedagógicamente competente. En este sentido, (Almerich et al., 2016) considera que para integrar las TIC en los procesos de enseñanza y aprendizaje es necesario poseer habilidades tecnológicas, pedagógicas y disciplinares, las cuales se encuentran propuestas también dentro del modelo de Conocimiento Técnico Pedagógico del Contenido (TPACK, *Technological Pedagogical Content Knowledge*), el cual es un modelo de extensión del conocimiento del contenido pedagógico, al cual se le ha adicionado el contenido tecnológico (Sandí-Delgado et al., 2022; Sandí-Delgado & Sanz, 2018a).

Finalmente, con base en los indicadores de la Tabla 9. 3, se plantean los siguientes objetivos del juego:

#### **a) Objetivos generales**

- ✓ Desarrollar destrezas y conocimientos en el área de redes de datos en el profesorado a través del uso de SkillNet para la adquisición de nuevas habilidades en tecnologías digitales.
- ✓ Enseñar al profesorado sobre el autoaprendizaje por medio de la TIC, con el propósito de fomentarles el aprendizaje autónomo.

#### **b) Objetivos específicos**

- ✓ Seleccionar las actividades de aprendizaje más apropiadas sobre el área de redes de datos para la mediación en el uso de SkillNet con el propósito de adecuarlas al perfil del profesorado participante.
- ✓ Implementar estrategias pedagógicas y de gestión de redes de datos que sirvan de guía para el profesorado mediante el uso de SkillNet para la adquisición de nuevas destrezas en tecnologías digitales.
- ✓ Diseñar actividades pedagógicas para la mediación por medio de diferentes formatos interactivos con el fin de desarrollar las competencias tecnológicas de autoaprendizaje.
- ✓ Definir criterios de evaluación diagnóstica y formativa sobre la autonomía de aprendizaje por parte del profesorado mediante herramientas TIC con el fin de medirles el nivel de competencias alcanzadas.

#### **FM&A03      Seleccionar el tipo de aplicación**

Para el diseño del prototipo no funcional de SkillNet, se ha elegido el tipo de aplicación Web para facilitar el funcionamiento compartido y compatible en distintas plataformas, el hecho de que sea una aplicación web amplía el foco de alcance o cobertura a la cual puede llegar a impactar el proyecto, máxime que, en términos de ejecución, cualquier dispositivo con acceso a internet, pantalla y cualquier navegador web puede acceder al juego (Armendariz et al., 2015; Carrión et al., 2019; Sandí-Delgado & Bazán, 2019, 2020).

##### **9.2.2. Fase 2. Lenguaje de programación**

En relación con el paradigma y tipo de lenguaje de programación utilizado para el diseño del prototipo no funcional del juego serio, se tomaron las siguientes decisiones:

#### **FM&A04      Paradigma de programación**

Para el diseño del prototipo no funcional de SkillNet, se eligió el paradigma de la Programación Orientada a Objetos, ya que permite que piezas de código se reutilicen e intercambien entre distintos programas (Ducasse & Pollet, 2017; Lafore, 2001).

#### **FM&A05      Tipo de lenguaje de programación**

El prototipo no funcional de SkillNet utilizará JavaScript para el Backend. Igualmente, se utilizará la biblioteca de JavaScript para construir la interfaz de usuario interactiva y para el manejo de componentes. Para brindar estilos a los componentes se utilizará CSS.

#### **9.2.3. Fase 3. Procesos de software**

Este apartado describe en detalle el proceso de software definido para guiar el diseño del prototipo de SkillNet. Para dicho diseño se utilizó una modelo de desarrollo denominada Agile, ya que esta metodología permite dotar al proyecto de flexibilidad a la hora de trabajar y para organizarse (Canós et al., 2003). En esta metodología el proyecto se traza en tareas que se tienen que ir completando para ir escalando. Unas de las metodologías más implementadas en este tipo de modelos de desarrollos es Scrum, especialmente utilizada en aquellos proyectos que permiten un cambio rápido de requisitos, este modelo es iterativo donde se toma retroalimentación de iteraciones pasadas (Canós et al., 2003).

#### **FM&A06      Modelo de desarrollo de software**

Para el desarrollo de la pieza de software se eligió trabajar bajo un modelo de desarrollo de prototipos, ya que el propósito de este es crear una parte no funcional de la aplicación (Lozano, 2017), en este caso en específico, el juego serio SkillNet.

#### **FM&A07      Diseño centrado en el usuario**

Para trabajar con el diseño centrado en el usuario, se decidió que, a nivel de interfaz, se trabajaría con las Hojas de Estilo en Cascada (CSS, *Cascading Style Sheets*), debido a que son un tipo de lenguaje que permite manejar el diseño y presentación de una página web, es decir, es lo que el usuario final puede observar

de forma gráfica (tipografía, color, tamaño, entre otros), separados de los que configuran el contenido. Igualmente, CSS opera en conjunto con el lenguaje HTML el cual es el encargado del contenido básico de las aplicaciones web.

Para una mejor exemplificación del diseño UX se detallan una serie de plantillas con el diseño del prototipo para esto se tomó la decisión de utilizar AdobeXD el cual es una herramienta de software comercial y profesional (requiere suscripción bajo pago), para el diseño de las plantillas requeridas. Para la creación de diseños del prototipo, se recomienda utilizar herramientas de software existentes o disponibles en el mercado, dependiendo de la organización y nivel económico, puede hacer uso software bajo licenciamiento con carácter de pago, en este sentido, para el diseño de plantillas se podría utilizar AdobeXD<sup>32</sup>, Figma<sup>33</sup>, Axure<sup>34</sup>, entre otras, las cuales son conocidas como las más comerciales y profesionales. De igual manera, se podría considerar la utilización de herramientas gratuitas web, pero conscientes de que pueden presentar ciertas limitaciones a nivel de diseño (Hussein & Ravna, 2015; Melgarejo-Torralba et al., 2022; Mohr et al., 2018). Para el caso en específico de SkillNet, se trabajará con AdobeXD.

#### 9.2.4. Fase 4. Ingeniería de requerimientos

En relación con los requerimientos de usuario, se parte del hecho de la existencia o manifestación previa del profesorado en relación con los requerimientos, mismos que fueron “levantados” o registrados a través de entrevistas, aplicación de cuestionarios y grupos focales realizados en el proceso de la tesis de magister del tesista (Sandí-Delgado et al., 2018), donde se solicitó explícitamente trabajar en el diseño e implementación de juegos serios dirigidos exclusivamente a la formación del profesorado en competencias tecnológicas acorde con las dimensiones e indicadores existentes, mismos que se abordan y describen en detalle en la fase siguiente.

#### FM&A08 Requerimientos funcionales y no funcionales

<sup>32</sup> <https://www.adobe.com/la/products/xd.html>

<sup>33</sup> <https://www.figma.com/>

<sup>34</sup> <https://www.axure.com/>

Para el prototipo no funcional de SkillNet se han identificado los siguientes requerimientos funcionales y no funcionales.

#### **9.2.4.1 Requerimientos funcionales**

En relación con los requerimientos funcionales del prototipo del juego SkillNet, la Tabla 9. 4 resume las Especificación de Requerimientos Funcionales del **Módulo Registro RF-1**.

**Tabla 9. 4 Módulo de Registro RF-1**

REF <sup>1</sup>	Función o subfunción <sup>2</sup>	Prioridad <sup>3</sup> y tipo <sup>4</sup>
RF-1.1	<p>El usuario debe estar registrado para poder acceder al juego serio, para ello debe llenar todos los campos del formulario de registro:</p> <ol style="list-style-type: none"> <li>1. Nombre:             <ol style="list-style-type: none"> <li>1.1. En este campo el usuario solo puede escribir un nombre que esté compuesto por caracteres únicamente alfabéticos ya sea mayúsculas o minúsculas.</li> </ol> </li> <li>2. Apellido:             <ol style="list-style-type: none"> <li>2.1. En este campo el usuario solo puede escribir un apellido que esté compuesto por caracteres únicamente alfabéticos ya sea mayúsculas o minúsculas.</li> </ol> </li> <li>3. Edad:             <ol style="list-style-type: none"> <li>3.1. En este campo el usuario debe seleccionar la fecha de nacimiento obtenida de un calendario.</li> </ol> </li> <li>4. Genero:             <ol style="list-style-type: none"> <li>4.1. El usuario debe elegir una opción de género (masculino, femenino o no específica).</li> </ol> </li> <li>5. País:             <ol style="list-style-type: none"> <li>5.1. El usuario deberá seleccionar un país del catálogo de países (combo box) que se desplegará en esta opción.</li> </ol> </li> <li>6. Correo electrónico:             <ol style="list-style-type: none"> <li>6.1. El usuario debe ingresar un correo que cumpla con un formato válido (correo@domain). Puede contener números y otros caracteres.</li> </ol> </li> <li>7. Contraseña:             <ol style="list-style-type: none"> <li>7.1. El usuario puede ingresar cualquier tipo de carácter en este campo para crear su contraseña.</li> <li>7.2. Debe tener un mínimo de 8 caracteres o un máximo de 30 caracteres.</li> </ol> </li> <li>8. Confirmar contraseña:             <ol style="list-style-type: none"> <li>8.1. En este campo el usuario debe ingresar nuevamente la contraseña que escribió en el campo "Contraseña" para ser confirmada.</li> </ol> </li> </ol>	A, E
RF-1.2	<p>Se debe validar cada campo del formulario tanto a nivel de Frontend como de Backend.</p> <ol style="list-style-type: none"> <li>1. Datos vacíos.</li> <li>2. Datos incorrectos (mínimo o máximo de caracteres, espacio al inicio o al final de los datos ingresados).</li> <li>3. Caracteres no admitidos en dicho campo.</li> </ol> <p>Los datos ingresados no siguen un formato especificado.</p>	A, E
RF-1.3	El sistema debe indicarle al usuario el campo o los campos con datos incorrectos en caso de que aplique.	
RF-1.4	Si todos los datos ingresados por el usuario son correctos (y han sido validados) el usuario puede presionar el botón "Registrarse" y el sistema lo redirecciona a una vista informativa donde le indica que debe verificar la cuenta ingresando a un enlace enviado por el sistema al correo electrónico registrado.	

RF-1.5	El usuario solo podrá iniciar sesión (por primera vez) cuando haya verificado su cuenta ingresando en el enlace enviado por el sistema al correo electrónico registrado.	
--------	--	--

La Tabla 9. 5 resume las Especificación de Requerimientos Funcionales del Módulo **Autentificación RF-2**.

**Tabla 9. 5 Módulo Autentificación RF-2**

REF <sup>1</sup>	Función o subfunción <sup>2</sup>	Prioridad <sup>3</sup> y tipo <sup>4</sup>
RF-2.1	<p>Se deben verificar todos los campos del login (correo electrónico y contraseña):</p> <ol style="list-style-type: none"> <li>1. Correo electrónico:           <ol style="list-style-type: none"> <li>1.1. El correo ingresado en este campo se debe validar a nivel de base de datos para corroborar que coincidan con los datos almacenados en base de datos.</li> </ol> </li> <li>2. Contraseña.</li> </ol> <p>La contraseña se debe verificar y coincidir con los datos de usuario almacenados en la base de datos.</p>	A, E

La Tabla 9. 6 resume las Especificación de Requerimientos Funcionales del Módulo **Contraseña RF-3**.

**Tabla 9. 6 Módulo Contraseña RF-3**

REF <sup>1</sup>	Función o subfunción <sup>2</sup>	Prioridad <sup>3</sup> y tipo <sup>4</sup>
RF-3.1	<p>Si el usuario olvida su contraseña puede recuperarla desde la página principal de autentificación, en el botón “R” El usuario puede recuperar la contraseña en caso de olvido, para ello se debe validar que el correo este registrado en la base de datos:</p> <ol style="list-style-type: none"> <li>1. Correo electrónico:           <ol style="list-style-type: none"> <li>1.1. Debe tener un formato válido (correo@domain). Puede contener números y otros caracteres.</li> </ol> </li> </ol>	A, E
RF-3.2	El sistema al validar el correo electrónico enviará un token a dicho correo para validar el cambio de contraseña.	A, O
RF-3.3	<p>El usuario deberá ingresar al correo electrónico para abrir el enlace que el sistema le envió y así ser redirigido al formulario para ingresar una nueva contraseña.</p> <ol style="list-style-type: none"> <li>1. Contraseña:           <ol style="list-style-type: none"> <li>1.1. El usuario puede ingresar cualquier tipo de carácter en este campo para crear su contraseña.</li> </ol> </li> </ol>	A, E

	<p>1.2. Debe tener un mínimo de 8 caracteres o un máximo de 30 caracteres.</p> <p>2. Confirmar contraseña: En este campo el usuario debe ingresar nuevamente la contraseña que escribió en el campo “Contraseña” para ser confirmada.</p>	
--	---	--

La Tabla 9.7 resume las Especificación de Requerimientos Funcionales del Módulo de Menú Inicio del Juego RF-4.

**Tabla 9.7 Módulo de Menú Inicio del Juego RF-4**

REF <sup>1</sup>	Función o subfunción <sup>2</sup>	Prioridad <sup>3</sup> y tipo <sup>4</sup>
RF-4.1	<p>El sistema muestra al usuario las opciones disponibles antes de empezar a jugar. El usuario al presionar:</p> <ol style="list-style-type: none"> <li>1. Iniciar Juego.             <ol style="list-style-type: none"> <li>1.1. Inicia la partida en la etapa 1 (primer nivel)</li> </ol> </li> <li>2. Continuar último nivel desbloqueado (si está disponible).             <ol style="list-style-type: none"> <li>2.1. El usuario comienza la partida en el último nivel de la última etapa desbloqueada.</li> </ol> </li> <li>3. Abandonar nivel (si está disponible).             <ol style="list-style-type: none"> <li>3.1. El jugador podrá abandonar el nivel actual. Cuando abandone se mostrará el menú principal.</li> </ol> </li> <li>4. Reiniciar nivel (si está disponible).             <ol style="list-style-type: none"> <li>4.1. Se reinicia el nivel de la etapa en que se encuentra el jugador.</li> </ol> </li> <li>5. Seleccionar un nivel específico.             <ol style="list-style-type: none"> <li>5.1. El usuario puede iniciar la partida en un nivel de una etapa disponible.</li> </ol> </li> <li>6. Mi progreso.             <ol style="list-style-type: none"> <li>6.1. En esta opción el jugador puede ver el progreso en porcentaje de la etapa y nivel en el que se encuentra, así como las estrellas obtenidas tanto a nivel general (etapa), así también como los niveles que haya superado.</li> </ol> </li> <li>7. Ranking.             <ol style="list-style-type: none"> <li>7.1. Se listarán los primeros 100 jugadores de manera descendente o ascendente tomando como criterio las estrellas obtenidas por nivel + el tiempo de duración de la etapa.</li> <li>7.2. Se podrá filtrar no solo por nivel sino también por etapa y utilizando los mismos criterios, pero para este filtro se sumarán las estrellas obtenidas entre los niveles que componen la etapa y de igual manera los tiempos.</li> </ol> </li> <li>8. Cerrar sesión. El usuario cierra la sesión y el sistema lo lleva a la pantalla principal para que pueda iniciar sesión nuevamente si así lo desea.</li> </ol>	A, E

La Tabla 9. 8 resume las Especificación de Requerimientos Funcionales del **Módulo Etapa 1 RF-5**

**Tabla 9. 8 Módulo Etapa 1 RF-5**

REF <sup>1</sup>	Función o subfunción <sup>2</sup>	Prioridad <sup>3</sup> y tipo <sup>4</sup>
RF-5.1	<p>La etapa #1 estará compuesta de 5 niveles los cuales el sistema mostrará al jugador y este deberá ir desbloqueando progresivamente, empezando desde el nivel 1. Los niveles son los siguientes:</p> <ol style="list-style-type: none"> <li>1. Clases de red (nivel 1).</li> <li>2. Máscara por defecto (nivel 2).</li> <li>3. Máscara variable (nivel 3).</li> <li>4. Id de red y broadcast (nivel 4).</li> </ol> <p>Host, primer IP utilizable, último IP utilizable, salto, broadcast y clase de red en el mismo nivel (nivel 5).</p>	A, E
RF-5.1.1	Cuando el jugador empiece el nivel 1, el sistema mostrará dos pantallas con información acerca de clases de red A, B y C para que el jugador adquiera conocimientos asociados al nivel actual y pueda aplicarlos en el juego.	A, E
RF-5.1.2	La primera pantalla con información tendrá un botón “siguiente” que el jugador puede presionar para que el sistema muestre la siguiente pantalla con más información sobre las clases de red.	A, E
RF-5.1.3	La segunda pantalla con más información sobre las clases de red tendrá un botón “siguiente” que el jugador puede presionar para que el sistema muestre la siguiente pantalla la cual contiene las instrucciones para jugar el nivel actual. Esta pantalla con instrucciones acerca de cómo mover los recuadros	A, E

	también tendrá un botón “siguiente” el cual llevará al jugador a empezar a jugar el nivel 1.	
RF-5.1.4	En el nivel 1 el sistema mostrará 5 recuadros. En el primer recuadro se muestra un id de red y en el recuadro 2 se arrastra una de las 3 opciones (Clase A, Clase B o Clase C). Cuando el jugador de con la opción correcta, el sistema generará un nuevo id de red y se repite el proceso hasta completar los pasos (6) de dicho nivel.	A, E
RF-5.2.1	Cuando el jugador empiece el nivel 2, el sistema mostrará una pantalla con información acerca de máscaras por defecto para que el jugador adquiera conocimientos asociados al nivel actual y pueda aplicarlos en el juego.	A, E
RF-5.2.2	La primera pantalla con información tendrá un botón “siguiente” el cual llevará al jugador a empezar a jugar el nivel actual.	A, E
RF-5.2.3	En el nivel 2 el sistema mostrará 5 recuadros. En el primer recuadro se muestra un id de red y en el recuadro 2 se arrastra una de las 3 opciones (255.0.0.0, 255.255.0.0, 255.255.255.0). Cuando el jugador de con la opción correcta, el sistema generará un nuevo id de red y se repite el proceso hasta completar los pasos (6) de dicho nivel.	A, E
RF-5.3-1	Cuando el jugador empiece el nivel 3, el sistema mostrará una pantalla con información acerca de broadcast para que el jugador adquiera conocimientos asociados al nivel actual y pueda aplicarlos en el juego.	A, E
RF-5.3.2	La primera pantalla con información tendrá un botón “siguiente” el cual llevará al jugador a empezar a jugar el nivel actual.	A, E
RF-5.3-3	En el nivel 4 el sistema mostrará 8 recuadros. En el primer recuadro se muestra un id de red y en el recuadro 2 se arrastra una de las opciones disponibles. Cuando el jugador de con la	A, E

	opción correcta, el sistema generará un nuevo id de red y se repite el proceso hasta completar los pasos (6) de dicho nivel.	
RF-5.4.1	Cuando el jugador empiece el nivel 5, el sistema mostrará una pantalla con información acerca de host, primer IP utilizable, último IP utilizable, salto, broadcast y clase de red para que el jugador adquiera conocimientos asociados al nivel actual y pueda aplicarlos en el juego.	A, E
RF-5.4.2	La primera pantalla con información tendrá un botón “siguiente” el cual llevará al jugador a empezar a jugar el nivel actual.	A, E
RF-5.4.3	En el nivel 5 el sistema mostrará 10 recuadros. En el primer recuadro se muestra un id de red y en el recuadro 2 se arrastra una de las opciones disponibles. El proceso de arrastrar opciones se repite hasta completar los pasos (6) de dicho nivel.	A, E
RF-5.5	Los mensajes informativos que se muestran al acertar o desacertar se cerrarán automáticamente después de algunos segundos.	M, E
RF-5.5.1	Los mensajes informativos que muestra el sistema cuando acierta o desacierta tendrán un botón “Aceptar” para que el usuario pueda cerrar dicho mensaje.	B, E
RF-5.5.2	El usuario podrá colocar el cursor del mouse sobre los mensajes informativos para que estos no se cierren automáticamente.	A, E
RF-5.5.3	Cuando el jugador acierte, el sistema muestra un mensaje de felicitaciones para motivarle a continuar aprendiendo.	M, E
RF-5.5.4	Cuando el jugador se equivoque, el sistema muestra un mensaje de respuesta incorrecta con una pista o información para que el jugador tenga más claro cuál podría ser la respuesta correcta.	A, E

RF-5.6	Cuando el jugador se equivoque, el sistema descontará una de las tres oportunidades (vidas) que tiene el jugador para completar todos los niveles de la etapa actual. Este proceso se repetirá hasta que el jugador ya no tenga más oportunidades.	A, E
RF-5.7	El jugador podrá salirse del nivel que se encuentre y en cualquier momento de la partida seleccionando la opción del menú “Abandonar nivel”. El sistema mostrará un mensaje de confirmación para validar que el usuario está seguro de abandonar el nivel.	A, E
RF-5.8	El jugador podrá reiniciar el nivel accediendo al menú del juego y presionando “Reiniciar nivel”. El sistema mostrará un mensaje de confirmación para validar que el usuario está seguro de reiniciar el nivel.	A, E
RF-5.9	Cuando el jugador termina un nivel, el sistema mostrará un cuadro con información sobre el nivel completado, así como la cantidad de estrellas obtenidas y un botón “siguiente”, el cual el jugador debe presionar para continuar el siguiente nivel hasta completar la etapa con todos sus niveles.	A, E
RF-5.10	El sistema deberá reiniciar la etapa (empezar nuevamente desde el nivel 1) actual cada vez que el jugador pierde sus 3 intentos.	A, E

La Tabla 9. 9 resume las Especificación de Requerimientos Funcionales del **Módulo Etapa #2 RF-6**

**Tabla 9. 9 Módulo Etapa #2 RF-6**

REF <sup>1</sup>	Función o subfunción <sup>2</sup>	Prioridad <sup>3</sup> y tipo <sup>4</sup>
RF-6.1	La etapa #2 estará compuesta de 2 niveles los cuales el sistema mostrará al jugador y este deberá ir desbloqueando progresivamente, empezando desde el nivel 1. Los niveles son los siguientes: 1. Conectar dispositivos con el cable correcto (nivel 1) 2. Crear topología con dispositivos y cables (nivel 2).	A, E

RF-6.1.1	Cuando el jugador empiece el nivel 1, el sistema mostrará una pantalla con información acerca de los tipos de conexión (cableado) entre diferentes dispositivos para que el jugador adquiera conocimientos asociados al nivel actual y pueda aplicarlos en el juego.	A, E
RF-6.1.2	La primera pantalla con información tendrá un botón “siguiente” el cual llevará al jugador a empezar a jugar el nivel actual.	A, E
RF-6.1.3	En el nivel 1 el sistema mostrará 5 recuadros y 2 imágenes de los dispositivos a conectar. En el primer recuadro se debe arrastrar una de las 4 opciones que se muestran abajo con un tipo de cable específico. El proceso de arrastrar opciones se repite hasta completar los pasos (6) de dicho nivel.	A, E
RF-6.2.1	Cuando el jugador empiece el nivel 2, el sistema mostrará 2 pantalla con información. La primera pantalla acerca de los tipos de conexión (cableado) con diferentes dispositivos para que el jugador adquiera conocimientos asociados al nivel actual y pueda aplicarlos en el juego.	M, E
RF-6.2.2	La primera pantalla con información tendrá un botón “siguiente” el cual llevará al jugador a la segunda pantalla con información acerca de los dispositivos necesarios para completar la topología del nivel actual.	A, E
RF-6.2.3	La segunda pantalla con más información acerca de los dispositivos a conectar tendrá un botón “siguiente” que el jugador puede presionar para que el sistema muestre la siguiente pantalla la cual contiene las instrucciones para jugar el nivel actual. Esta pantalla con instrucciones acerca de cómo mover los recuadros también tendrá un botón “siguiente” el cual llevará al jugador a empezar a jugar el nivel 2.	A, E
RF-6.2.4	En el nivel 2 el sistema mostrará 11 recuadros. En el primer recuadro aparecerá el nombre de un dispositivo y el usuario deberá arrastrar la opción que corresponda a ese nombre. Entre las opciones para arrastrar habrá un rack, un patch panel, un router, un tester, cable UTP, cable coaxial, dispositivo final (laptop), una ponchadora, un conector RJ-45 y una prensadora.	A, E
RF-6.3	A la par del recuadro de respuesta habrá un botón tipo icono el cual el usuario puede presionar para obtener algún tipo de ayuda o pista para dar con la respuesta correcta.	M, E
RF-6.4	El usuario podrá colocar el cursor del mouse sobre los mensajes informativos para que estos no se cierren automáticamente.	A, E
RF-6.4.1	Cuando el jugador acierte, el sistema muestra un mensaje de felicitaciones para motivarle a continuar aprendiendo.	M, E
RF-6.4.2	Cuando el jugador se equivoque, el sistema muestra un mensaje de respuesta incorrecta con una pista o información para que el jugador tenga más claro cuál podría ser la respuesta correcta.	A, E
RF-6.5	Cuando el jugador se equivoque, el sistema descontará una de las tres oportunidades (vidas) que tiene el jugador para completar todos los niveles de la etapa actual. Este proceso se repetirá hasta que el jugador ya no tenga más oportunidades.	A, E

RF-6.6	El jugador podrá salirse del nivel que se encuentre y en cualquier momento de la partida seleccionando la opción del menú “Abandonar nivel”. El sistema mostrará un mensaje de confirmación para validar que el usuario está seguro de abandonar el nivel.	A, E
RF-6.7	El jugador podrá reiniciar el nivel accediendo al menú del juego y presionando “Reiniciar nivel”. El sistema mostrará un mensaje de confirmación para validar que el usuario está seguro de reiniciar el nivel.	A, E
RF-6.8	Cuando el jugador termina un nivel, el sistema mostrará un cuadro con información sobre el nivel completado, así como la cantidad de estrellas obtenidas y un botón “siguiente”, el cual el jugador debe presionar para continuar el siguiente nivel hasta completar la etapa con todos sus niveles.	A, E
RF-6.9	El sistema deberá reiniciar la etapa actual (empezar nuevamente desde el nivel 1) cada vez que el jugador pierde sus 3 intentos.	A, E

La Tabla 9. 10 resume las Especificación de Requerimientos Funcionales del **Módulo Etapa #3 RF-7**

**Tabla 9. 10 Módulo Etapa #3 RF-7**

<b>REF<sup>1</sup></b>	<b>Función o subfunción<sup>2</sup></b>	<b>Prioridad<sup>3</sup> y tipo<sup>4</sup></b>
RF-7.1	La etapa #3 estará compuesta de un nivel el cual el sistema mostrará al jugador. 1. Seleccionar digitar el comando correcto según el tipo de configuración seleccionada.	A, E
RF-7.1.1	Cuando el jugador empiece el nivel 1, el sistema mostrará una pantalla con información acerca de los objetivos a cumplir en el nivel, esta tendrá un botón “siguiente” el cual llevará al jugador a empezar a jugar el nivel actual.	A, E
RF-7.1.2	Seguidamente se le mostrará al usuario una serie de opciones a seleccionar que contendrán diferentes protocolos de red o dispositivos de red a configurar.	A, E
RF-7.1.3	Para la opción seleccionada, se desplegará una lista de comandos de distintos protocolos y configuraciones los cuales interactúan con el componente que validará las opciones elegidas.	A, E
RF-7.2	Cuando el jugador acierte, el sistema muestra un mensaje de felicitaciones para motivarle a continuar aprendiendo.	M, E
RF-7.2.1	El usuario podrá colocar el cursor del mouse sobre los mensajes informativos para que estos no se cierren automáticamente.	A, E
RF-7.3	A la par del recuadro de respuesta habrá un botón tipo ícono el cual el usuario puede presionar para obtener algún tipo de ayuda o pista para dar con la respuesta correcta.	M, E
RF-7.4	Cuando el jugador se equivoque, el sistema muestra un mensaje de respuesta incorrecta con una pista o información para que el jugador tenga más claro cuál podría ser la respuesta correcta.	A, E
RF-7.5	Cuando el jugador se equivoque, el sistema descontará una de las tres oportunidades (vidas) que tiene el jugador para completar todos los niveles de la etapa actual. Este proceso se repetirá hasta que el jugador ya no tenga más oportunidades.	A, E
RF-7.6	El jugador podrá salirse del nivel que se encuentre y en cualquier momento de la partida seleccionando la opción del menú “Abandonar nivel”. El sistema mostrará un mensaje de confirmación para validar que el usuario está seguro de abandonar el nivel.	A, E
RF-7.7	El jugador podrá reiniciar el nivel accediendo al menú del juego y presionando “Reiniciar nivel”. El sistema mostrará un mensaje de confirmación para validar que el usuario está seguro de reiniciar el nivel.	A, E
RF-7.8	Cuando el jugador termina un nivel, el sistema mostrará un cuadro con información sobre el nivel completado, así como la cantidad de estrellas obtenidas y un botón “siguiente”, el cual el jugador debe presionar para continuar el siguiente nivel hasta completar la etapa con todos sus niveles.	A, E
RF-7.9	El sistema deberá reiniciar la etapa actual (empezar nuevamente desde el nivel 1) cada vez que el jugador pierde sus 3 intentos.	A, E

<sup>1</sup> Indica nivel de la función, por ejemplo 1.1, 1.2, 1.2.1, etc.

2 Niveles de detalle para las funciones. 3 Prioridad = A (alta), M (media) o B (baja).  
4 Tipo = E (evidente al usuario) u O (oculto al usuario pero evidente para el analista).

### 9.2.4.2 Requerimientos no funcionales

En relación con los requerimientos no funcionales del prototipo del juego SkillNet, la Tabla 9. 11 resume las Especificación de Requerimientos No Funcionales (RNF) y comunes entre RNF.

**Tabla 9. 11 Especificación de Requerimientos No Funcionales y comunes RNF**

RNF <sup>1</sup>	Función o subfunción <sup>2</sup>	Prioridad <sup>3</sup>
RNF-1	<b>Fiabilidad/Confiabilidad.</b> 1. Disponibilidad.	A
RNF-2	<b>Portabilidad.</b> 1. Adaptabilidad.	A
RNF-3	<b>Seguridad.</b> 1. Confidencialidad. 2. Integridad. 3. Autenticidad. 4. Responsabilidad.	A
RNF-4	<b>Mantenibilidad.</b> 1. Modularidad. 2. Reusabilidad. 3. Capacidad de ser modificado.	M
RNF-5	<b>Usabilidad.</b> 1. Aprendizaje. 2. Operabilidad. 3. Protección frente a errores de usuario. 4. Estética.	A
RNF-5	<b>Eficiencia.</b> 1. Tiempo de respuesta (¿Cómo podemos validar el tiempo de respuesta?) 2. Utilización de recursos. 3. Comportamiento en el tiempo. 4. Capacidad.	A

1 indica la función.

2 niveles de detalle para las funciones.

3 prioridad = A (alta), M (media) o B (baja).

**FM&A09 Casos de uso**

Para el prototipo no funcional de SkillNet se definieron diferentes casos de uso, tal como el login, solicitar y recuperar contraseñas. En la Tabla 9. 12 se puede apreciar el caso de uso para el **login** de la aplicación del juego.

**Tabla 9. 12 Caso de Uso Login**

Realizar Login		
Línea	Campo	Descripción
1	Id de caso	CU-01
2	Actores	Usuario
3	Tipo de caso uso	Primario
4	Breve descripción	Este caso de uso permite el acceso a los usuarios con permisos en la aplicación.
5	Precondiciones	El usuario tiene que encontrarse en la página de login administrativo.
6	Flujo de Eventos	01. El usuario ingresa un correo. 02. El sistema valida que el formato de correo sea válido (Regex Correo) 04. El usuario ingresa una contraseña. 05. El usuario da clic al botón de Acceder 06. El sistema verifica la validez de los datos ingresados 07. El sistema permite el acceso al usuario
7	Flujos alternativos	<b>El usuario ingresa un correo con formato inválido (Después del paso 2)</b> 01. El sistema alerta mediante una etiqueta bajo el input al usuario 02. Volver al paso 1 <b>El usuario ingresa un datos inválidos (Después del paso 5)</b> 01. El sistema alerta mediante un Toast al usuario 02. Volver al paso 1
8	Postcondiciones	01. El usuario tiene acceso a la aplicación al loguearse
9	Excepciones	<b>El sistema pierde la conexión al backend</b> 01. El sistema alerta al usuario de la excepción
10	Extensión	Ninguno
11	Inclusión	Ninguno
12	Prioridad	Alta
13	Reglas de negocio	BR-32, BR-33, BR-34
14	Frecuencia de uso	Frecuente
15	Requisitos especiales	Ninguno
16	Observaciones	Ninguno

En la Tabla 9. 13 se puede apreciar el caso de uso para la **solicitud de recuperación de contraseña** de la aplicación del juego.

**Tabla 9. 13 Caso de uso solicitar recuperación de contraseña**

Solicitar recuperación de contraseña		
Línea	Campo	Descripción
1	Id de caso	CU-02
2	Actores	Usuario
3	Tipo de caso uso	Primario
4	Breve descripción	Este caso permite al usuario solicitar un correo para la recuperación de la contraseña en caso de haberla olvidado
5	Precondiciones	Tener una cuenta en la página.
6	Flujo de Eventos	<p>01. El usuario ingresa su correo</p> <p>02. El sistema valida que el formato de correo sea válido (Regex Correo)</p> <p>03. El usuario da clic al botón de Recordar</p> <p>04. El sistema verifica la validez de los datos ingresados</p> <p>05. El sistema envía el correo de recuperación</p> <p>06. El sistema alerta mediante un toast al usuario</p> <p>07. El sistema dirige al usuario al login</p>
	Flujos alternativos	<b>El usuario ingresa un correo con formato inválido (Después del paso 2)</b>
		01. El sistema alerta mediante una etiqueta bajo el input al usuario
		02. Volver al paso 1
		<b>El usuario ingresa un correo que no ha sido registrado (Después del paso 2)</b>
		01. El sistema alerta mediante un toast al usuario
		02. Volver al paso 1
8	Postcondiciones	01. El usuario recibirá un correo para cambiar su contraseña
9	Excepciones	<b>El sistema pierde la conexión al backend</b>
10		01. El sistema alerta al usuario de la excepción
11	Extensión	Ninguno
12	Inclusión	Ninguno
13	Prioridad	Alta
14	Reglas de negocio	BR-33
15	Frecuencia de uso	Media
15	Requisitos especiales	Ninguno
16	Observaciones	Ninguno

En la Tabla 9. 14 se puede apreciar el caso de uso para **recuperación de contraseña** de la aplicación del juego.

**Tabla 9. 14 Caso de uso recuperación de contraseña**

Recuperar contraseña por correo		
Línea	Campo	Descripción
1	Id de caso	CU-03
2	Actores	Usuario
3	Tipo de caso uso	Primario
4	Breve descripción	Este caso permite al usuario cambiar la contraseña en caso de haberla olvidado
5	Precondiciones	Haber solicitado el correo de recuperación
6	Flujo de Eventos	<p>01. El usuario da clic al URL en el correo</p> <p>02. El sistema verifica el token de la URL</p> <p>03. El sistema muestra la ventana de cambio de contraseña</p> <p>04. El usuario ingresa la nueva contraseña</p> <p>05. El sistema valida el formato de la contraseña</p> <p>06. El usuario ingresa la confirmación de la nueva contraseña</p> <p>07. El sistema valida el formato de la confirmación y que sean iguales</p> <p>08. El usuario da clic en el botón de confirmar</p> <p>09. El sistema confirma la transacción guardando los datos</p> <p>10. El sistema alerta al usuario del cambio</p> <p>11. El sistema redirige al usuario a la página de login</p>
7	Flujos alternativos	<p><b>El token no es válido</b></p> <p>01. El sistema alerta al usuario</p> <p>02. El sistema redirige al usuario al inicio</p> <p><b>El usuario ingresa una contraseña con formato inválido (Después del paso 4)</b></p> <p>01. El sistema alerta mediante una etiqueta bajo el input al usuario</p> <p>02. Volver al paso 4</p> <p><b>El usuario ingresa una confirmación de contraseña inválida (Después del paso 4)</b></p> <p>01. El sistema alerta mediante una etiqueta bajo el input al usuario</p> <p>02. Volver al paso 6</p>
8	Postcondiciones	01. El usuario habrá cambiado su contraseña
9	Excepciones	<p><b>El sistema pierde la conexión al backend</b></p> <p>01. El sistema alerta al usuario de la excepción</p> <p><b>El token expiró</b></p> <p>01. El sistema alerta al usuario de la excepción y redirige al inicio</p>
10	Extensión	Ninguno
11	Inclusión	Ninguno
12	Prioridad	Alta
13	Reglas de negocio	BR-33
14	Frecuencia de uso	Media
15	Requisitos especiales	N/A
16	Observaciones	El correo de recuperación tiene una vigencia de 24 horas

En la Tabla 9. 15 se puede apreciar el caso de uso para **registro de usuario** a la aplicación del juego.

**Tabla 9. 15 Caso de uso registro de usuario**

Registro de Usuario		
Línea	Campo	Descripción
1	Id de caso	CU-04
2	Actores	Usuario
3	Tipo de caso uso	Primario
4	Breve descripción	Una persona accede al módulo de registro de la aplicación web, para crear su usuario
5	Precondiciones	El usuario deberá de dirigirse al sitio web
6	Flujo de Eventos	Registrarse: 01. El usuario selecciona la opción de registrarse. 02. Se despliega un formulario para ingresar los datos del usuario. 03. Se rellenan los campos del formulario con la información solicitada: nombre, apellido, edad, país, género, correo, contraseña 04. Comprobar que los datos estén correctamente agregados. 05. Selecciona registrarse
		<b>Usuario existente (después del paso 05 F.E)</b>
7		01. Se manda una alerta de que ya está registrado. 02. Se redirige a la página principal.
		<b>Campos vacíos o inválidos (después del paso 05 F.E)</b>
		01. Se muestra una alerta campos vacíos o inválidos. 02. Se indican los campos vacíos o inválidos. 03. Verificar los campos rellenados.
8	Postcondiciones	01. El sistema se actualiza con el usuario registrado. 02. Para ingresar debe ingresar los datos de usuario. 03. El sistema notifica que se registró exitosamente.
9	Excepciones	<b>Datos Incorrectos</b> 01 Si hay datos incorrectos el formulario no se envía.
10	Extensión	Ninguno
11	Inclusión	Ninguno
12	Prioridad	Alta
13	Reglas de negocio	BR-1, BR-3, BR-4
14	Frecuencia de uso	Frecuente
15	Requisitos especiales	Ninguno
16	Observaciones	Ninguno

En la Tabla 9. 16 se puede apreciar el caso de uso **etapa 1 – nivel 1 al 5** del juego SkillNet.

**Tabla 9. 16 Caso de uso etapa 1 - nivel 1 al 5 del juego**

Jugar etapa #1 nivel 1 al 5		
Línea	Campo	Descripción
1	Id de caso	CU-05
2	Actores	Usuario
3	Tipo de caso uso	Primario
4	Breve descripción	Este caso permite al usuario jugar los niveles del 1 al 5 de la etapa #1
5	Precondiciones	El usuario debe estar registrado y haber iniciado sesión.
6	Flujo de Eventos	01. El usuario da click en el botón del menú "Iniciar Juego".

		02. El sistema muestra una pantalla con información sobre el nivel actual. 03. El usuario da click en el botón "siguiente" en la pantalla de información sobre el nivel actual 04. El sistema muestra una pantalla con las instrucciones sobre el juego. 05. El usuario da click en el botón "siguiente" en la pantalla con las instrucciones sobre el juego. 06. El sistema comienza el juego. 07. El usuario arrastra la opción correcta al recuadro de respuesta. 08. El sistema muestra al usuario un mensaje de felicitaciones para motivarle a continuar aprendiendo. 09. El usuario repite el paso 07 hasta completar los 6 pasos del nivel actual satisfactoriamente (completa el nivel). 10. El sistema muestra una ventana con un mensaje y las estrellas obtenidas en el nivel actual.
7	Flujos alternativos	<b>El usuario se equivoca 1 vez en la respuesta (Después del paso 06)</b> 01. El usuario arrastra una opción incorrecta al recuadro de respuesta. 02. El sistema muestra un mensaje de error con una pista para que el usuario pueda determinar la respuesta correcta. 03. El usuario arrastra la opción correcta al recuadro de respuesta. 04. El sistema muestra al usuario un mensaje de felicitaciones para motivarle a continuar aprendiendo. 05. El usuario repite el paso 03 hasta completar los 6 pasos del nivel actual satisfactoriamente (completa el nivel). 06. El sistema muestra una ventana con un mensaje y las estrellas obtenidas (2) en el nivel actual.
		<b>El usuario se equivoca 2 veces en la respuesta (Después del paso 06)</b> 01. El usuario arrastra una opción incorrecta al recuadro de respuesta 2 veces. 02. El sistema muestra un mensaje de error con una pista para que el usuario pueda determinar la respuesta correcta. 03. El usuario arrastra la opción correcta al recuadro de respuesta. 04. El sistema muestra al usuario un mensaje de felicitaciones para motivarle a continuar aprendiendo. 05. El usuario repite el paso 03 hasta completar los 6 pasos del nivel actual satisfactoriamente (completa el nivel). 06. El sistema muestra una ventana con un mensaje y las estrellas obtenidas (1) en el nivel actual.
		<b>El usuario se equivoca 3 veces en la respuesta (Después del paso 06)</b> 01. El usuario arrastra una opción incorrecta al recuadro de respuesta 3 veces. 02. El sistema advierte al usuario que perdió sus 3 intentos y le devuelve al nivel 1 de la etapa actual (reiniciar etapa desde cero).
8	Postcondiciones	01. El sistema redirige al jugador al nivel o etapa correspondiente.
9	Excepciones	01. El usuario abandona el nivel actual seleccionando una de las opciones del menú.
10	Extensión	-
11	Inclusión	-
12	Prioridad	Alta
13	Reglas de negocio	BR-11, BR-12, BR-13, BR-14, BR-15, BR-16, BR-17, BR-18, BR-19, BR-20, BR-21, BR-22, BR-23, BR-24, BR-25, BR-26
14	Frecuencia de uso	Frecuente

15	Requisitos especiales	Ninguno
16	Observaciones	Ninguno

En la Tabla 9. 17 se puede apreciar el caso de uso **etapa 2 – nivel 1 al 2** del juego SkillNet.

**Tabla 9. 17 Caso de uso etapa 2 - nivel 1 al 2 del juego**

<b>Jugar etapa #2 nivel 1 al 2</b>		
<b>Línea</b>	<b>Campo</b>	<b>Descripción</b>
1	Id de caso	CU-06
2	Actores	Usuario
3	Tipo de caso uso	Primario
4	Breve descripción	Este caso permite al usuario jugar los niveles 1 y 2 de la etapa #2
5	Precondiciones	01. El usuario debe estar registrado y haber iniciado sesión. 02. El usuario debe haber completado la etapa #1.
6	Flujo de Eventos	01. El usuario da click en el botón del menú "Iniciar Juego". 02. El sistema muestra una pantalla con información sobre el nivel actual. 03. El usuario da click en el botón "siguiente" en la pantalla de información sobre el nivel actual 04. El sistema muestra una pantalla con las instrucciones sobre el juego. 05. El usuario da click en el botón "siguiente" en la pantalla con las instrucciones sobre el juego. 06. El sistema comienza el juego. 07. El usuario arrastra la opción correcta al recuadro de respuesta. 08. El sistema muestra al usuario un mensaje de felicitaciones para motivarle a continuar aprendiendo. 09. El usuario repite el paso 07 hasta completar los 6 pasos del nivel actual satisfactoriamente (completa el nivel). 10. El sistema muestra una ventana con un mensaje y las estrellas obtenidas en el nivel actual.
7	Flujos alternativos	<b>El usuario se equivoca 1 vez en la respuesta (Después del paso 06)</b> 01. El usuario arrastra una opción incorrecta al recuadro de respuesta. 02. El sistema muestra un mensaje de error con una pista para que el usuario pueda determinar la respuesta correcta. 03. El usuario arrastra la opción correcta al recuadro de respuesta. 04. El sistema muestra al usuario un mensaje de felicitaciones para motivarle a continuar aprendiendo. 05. El usuario repite el paso 03 hasta completar los 6 pasos del nivel actual satisfactoriamente (completa el nivel). 06. El sistema muestra una ventana con un mensaje y las estrellas obtenidas (2) en el nivel actual. <b>El usuario se equivoca 2 veces en la respuesta (Después del paso 06)</b> 01. El usuario arrastra una opción incorrecta al recuadro de respuesta 2 veces. 02. El sistema muestra un mensaje de error con una pista para que el usuario pueda determinar la respuesta correcta. 03. El usuario arrastra la opción correcta al recuadro de respuesta.

		04. El sistema muestra al usuario un mensaje de felicitaciones para motivarle a continuar aprendiendo. 05. El usuario repite el paso 03 hasta completar los 6 pasos del nivel actual satisfactoriamente (completa el nivel). 06. El sistema muestra una ventana con un mensaje y las estrellas obtenidas (1) en el nivel actual. <b>El usuario se equivoca 3 veces en la respuesta (Después del paso 06)</b> 01. El usuario arrastra una opción incorrecta al recuadro de respuesta 3 veces. 02. El sistema advierte al usuario que perdió sus 3 intentos y le devuelve al nivel 1 de la etapa actual (reiniciar etapa desde cero).
8	Postcondiciones	01. El sistema redirige al jugador al nivel o etapa correspondiente.
9	Excepciones	01. El usuario abandona el nivel actual seleccionando una de las opciones del menú.
10	Extensión	Ninguno
11	Inclusión	Ninguno
12	Prioridad	Alta
13	Reglas de negocio	BR-11, BR-12, BR-13, BR-14, BR-15, BR-16, BR-17, BR-18, BR-19, BR-20, BR-21, BR-22, BR-23, BR-24, BR-25, BR-26
14	Frecuencia de uso	Frecuente
15	Requisitos especiales	Ninguno
16	Observaciones	Ninguno

En la Tabla 9. 18 se puede apreciar el caso de uso **etapa 3 – nivel 1** del juego SkillNet.

**Tabla 9. 18 Caso de uso etapa 3 – nivel 1**

Jugar etapa #3 nivel 1		
Línea	Campo	Descripción
1	Id de caso	CU-07
2	Actores	Usuario
3	Tipo de caso uso	Primario
4	Breve descripción	Este caso permite al usuario jugar el nivel 1 de la etapa #3
5	Precondiciones	El usuario debe estar registrado y haber iniciado sesión.
6	Flujo de Eventos	01. El usuario da click en el botón del menú "Iniciar Juego". 02. El sistema muestra una pantalla con información sobre el nivel actual. 03. El usuario da click en el botón "siguiente" en la pantalla de información sobre el nivel actual 04. El sistema muestra una pantalla para que el usuario seleccione una de las 6 opciones. 05. El jugador selecciona una opción de las que se muestran en pantalla. 06. El sistema muestra una pantalla con las instrucciones del juego. 07. El usuario da click en el botón "siguiente" en la pantalla con las instrucciones del juego. 08. El sistema comienza el juego. 09. El sistema muestra 2 recuadros; 1 con una lista posibles comandos relacionados al protocolo elegido y otro recuadro donde debe escribir los comandos que le parezcan pertinentes.

		10. El usuario escribe correctamente cada uno de los comandos y completa el nivel satisfactoriamente. 11. El sistema muestra un mensaje de felicitaciones por cada respuesta correcta. 12. El sistema muestra una ventana con un mensaje y las estrellas obtenidas en el nivel actual.
7	Flujos alternativos	<b>El usuario se equivoca 1 vez en la respuesta (Después del paso 08)</b> 01. El usuario escribe un comando inadecuado respecto al protocolo seleccionado. 02. El sistema muestra un mensaje de error con una pista para que el usuario pueda determinar un comando correcto. 03. El usuario escribe correctamente cada uno de los comandos y completa el nivel satisfactoriamente. 04. El sistema muestra al usuario un mensaje de felicitaciones para motivarle a continuar aprendiendo. 06. El sistema muestra una ventana con un mensaje y las estrellas obtenidas (2) en el nivel actual. <b>El usuario se equivoca 2 veces en la respuesta (Después del paso 08)</b> 01. El usuario escribe equivocadamente 2 veces un comando respecto al protocolo seleccionado. 02. El sistema muestra un mensaje de error con una pista para que el usuario pueda determinar un comando correcto. 03. El usuario escribe correctamente cada uno de los comandos y completa el nivel satisfactoriamente. 04. El sistema muestra al usuario un mensaje de felicitaciones para motivarle a continuar aprendiendo. 06. El sistema muestra una ventana con un mensaje y las estrellas obtenidas (1) en el nivel actual. <b>El usuario se equivoca 3 veces en la respuesta. (Después del paso 08)</b> 01. El usuario escribe equivocadamente 3 veces un comando respecto al protocolo seleccionado. 02. El sistema advierte al usuario que perdió sus 3 intentos y le devuelve al nivel 1 de la etapa actual (reiniciar etapa desde cero).
8	Postcondiciones	01. El sistema redirige al jugador al nivel o etapa correspondiente.
9	Excepciones	01. El usuario abandona el nivel actual seleccionando una de las opciones del menú.
10	Extensión	Ninguno
11	Inclusión	Ninguno
12	Prioridad	Alta
13	Reglas de negocio	BR-11, BR-12, BR-13, BR-14, BR-15, BR-16, BR-17, BR-18, BR-19, BR-20, BR-21, BR-22, BR-23, BR-24, BR-25, BR-26
14	Frecuencia de uso	Frecuente
15	Requisitos especiales	Ninguno
16	Observaciones	Ninguno

En la Tabla 9. 19 se puede apreciar el caso de uso **agregar usuario al grupo** del juego.

**Tabla 9. 19 Caso de uso agregar usuario al grupo**

Crear Grupo		
Línea	Campo	Descripción
1	Id de caso	CU-08
2	Actores	Usuario
3	Tipo de caso uso	Secundario
4	Breve descripción	El usuario registrado podrá crear grupos donde puede agregar otros usuarios para ver sus métricas
5	Precondiciones	01. El usuario debe estar registrado y haber iniciado sesión.
6	Flujo de Eventos	<b>Agregar Grupo</b> 01. El usuario accede a sus estadísticas donde se presentará una opción de agregar grupo 02. El usuario selecciona la opción de agregar grupo 03. Se despliega una vista para la creación de grupos donde el usuario agrega un nombre y crear su grupo
		<b>Grupo existente (después del paso 03 F.E)</b> 01. Se manda una alerta de que ya está registrado el grupo.
		<b>Campos vacíos o inválidos (después del paso 03 F.E)</b> 01. Se muestra una alerta campos vacíos o inválidos. 02. Se indican los campos vacíos o inválidos. 03. Verificar los campos llenados.
		01. El sistema se actualiza con el grupo registrado 02. El sistema notifica que se registró exitosamente.
9	Excepciones	Ninguno
10	Extensión	Ninguno
11	Inclusión	Ninguno
12	Prioridad	Ninguno
13	Reglas de negocio	BR-30
14	Frecuencia de uso	Media
15	Requisitos especiales	Ninguno
16	Observaciones	01 - El usuario a agregar al grupo debe estar previamente registrado, de lo contrario se envía una invitación para su posterior registro.

En la Tabla 9. 20 se puede apreciar el caso de uso **eliminar grupo** del juego.

**Tabla 9. 20 Caso de uso eliminar grupo**

Eliminar Grupo		
Línea	Campo	Descripción
1	Id de caso	CU-09
2	Actores	Usuario
3	Tipo de caso uso	Secundario
4	Breve descripción	El usuario registrado podrá eliminar los grupos que haya creado
5	Precondiciones	01. El usuario debe estar registrado y haber iniciado sesión.
6	Flujo de Eventos	<b>Eliminar Grupo</b> 01. El usuario accede al módulo de grupos 02. El usuario visualiza todos los grupos registrados 03. El usuario selecciona la opción de eliminar grupo 04. Se despliega una ventana informativa para confirmar la operación 05. El usuario confirma la operación.
		<b>Grupo existente (después del paso 03 F.E)</b> 01. Se manda una alerta de que ya está registrado el grupo.
		<b>Campos vacíos o inválidos (después del paso 03 F.E)</b>
7	Flujos alternativos	

		01. Se muestra una alerta campos vacíos o inválidos. 02. Se indican los campos vacíos o inválidos. 03. Verificar los campos rellenados.
8	Postcondiciones	01. El sistema se actualiza con el grupo registrado 02. El sistema notifica que se registró exitosamente.
9	Excepciones	Ninguno
10	Extensión	Ninguno
11	Inclusión	Ninguno
12	Prioridad	Ninguno
13	Reglas de negocio	BR-31
14	Frecuencia de uso	Media
15	Requisitos especiales	Ninguno
16	Observaciones	01 - El usuario a agregar al grupo debe estar previamente registrado, de lo contrario se envía una invitación para su posterior registro.

En la Tabla 9. 21 se puede apreciar el caso de uso **agregar usuario al grupo** del juego.

**Tabla 9. 21 Caso de uso agregar usuario al grupo**

<b>Agregar usuario a grupo</b>		
<b>Línea</b>	<b>Campo</b>	<b>Descripción</b>
1	Id de caso	CU-10
2	Actores	Usuario
3	Tipo de caso uso	Secundario
4	Breve descripción	El usuario registrado podrá agregar otros usuarios a sus grupos
5	Precondiciones	01. El usuario debe estar registrado y haber iniciado sesión.
6		<b>Agregar usuario</b>
		01. El usuario accede al módulo de agregar usuario.
		02. El usuario ingrese un correo electrónico en el input correspondiente
6	Flujo de Eventos	03. El usuario selecciona enviar invitación.
7		<b>Usuario existente (después del paso 05 F.E)</b>
		01. Se manda una alerta de que ya está registrado.
		02. Se redirige a la página principal.
7		<b>Campos vacíos o inválidos (después del paso 05 F.E)</b>
		01. Se muestra una alerta campos vacíos o inválidos.
		02. Se indican los campos vacíos o inválidos.
7	Flujos alternativos	03. Verificar los campos rellenados.
8	Postcondiciones	01. El sistema se actualiza con el usuario agregado. 02. El sistema notifica que se registró exitosamente.
9	Excepciones	Ninguno
10	Extensión	Ninguno
11	Inclusión	Ninguno
12	Prioridad	Ninguno
13	Reglas de negocio	BR-29
14	Frecuencia de uso	Media
15	Requisitos especiales	Ninguno
16	Observaciones	Ninguno

En la Tabla 9. 22 se puede apreciar el caso de uso **eliminar usuario al grupo**.

**Tabla 9. 22 Caso de uso eliminar usuario al grupo**

Eliminar usuario de un grupo		
Línea	Campo	Descripción
1	Id de caso	CU-11
2	Actores	Usuario
3	Tipo de caso uso	Secundario
4	Breve descripción	El usuario registrado podrá eliminar usuarios de los grupos que administre.
5	Precondiciones	01. El usuario debe estar registrado y haber iniciado sesión. <b>Agregar usuario</b> 01. El usuario accede al módulo de grupos 02. El usuario selecciona el grupo a administrar. 03. El usuario visualiza una tabla con registros 04. El usuario selecciona el registro que desea eliminar 05. El usuario selecciona la acción eliminar 06. El sistema alerta sobre la eliminación 07. El usuario confirma la eliminación 08. El sistema alerta que el usuario ha sido eliminado
6	Flujo de Eventos	
7	Flujos alternativos	Ninguno
8	Postcondiciones	01. El sistema se actualiza con el usuario agregado. 02. El sistema notifica que se registró exitosamente.
9	Excepciones	Ninguno
10	Extensión	Ninguno
11	Inclusión	Ninguno
12	Prioridad	Ninguno
13	Reglas de negocio	BR-31
14	Frecuencia de uso	Media
15	Requisitos especiales	Ninguno
16	Observaciones	Ninguno

**FM&A10 Reglas de negocio**

Para el prototipo no funcional de SkillNet se han definido las siguientes **reglas de negocio** (BR):

**Tabla 9. 23 Reglas de negocio**

BR	Descripción	Módulo
BR-1	Cualquier persona puede registrarse.	Registro
BR-2	Una persona puede jugar solo si está registrado y con la sesión iniciada.	Registro
BR-3	La edad mínima para registrarse es de 12 años.	Registro
BR-4	La edad máxima para registrarse es de 100 años.	Registro
BR-5	Cualquier jugador registrado puede crear 0, 1 o muchos grupos.	General
BR-6	Un jugador puede pertenecer a múltiples grupos.	General

**CAPÍTULO IX. PROTOTIPO NO FUNCIONAL DEL JUEGO SERIO SKILLNET**

BR-7	Solo el jugador propietario del grupo puede ver las estadísticas y progresos de los miembros del grupo.	General
BR-8	Un jugador puede eliminar el o los grupos que ha creado.	General
BR-9	El jugador que ha creado un grupo puede invitar a otros jugadores a través de un enlace (link).	General
BR-10	Los jugadores pueden rechazar unirse a un grupo ignorando el enlace de invitación.	General
BR-11	El jugador debe completar el nivel en que se encuentre para poder recibir calificación con las estrellas (1, 2 o 3 estrellas).	Etapa 1, 2 y 3
BR-12	El jugador debe completar al menos una etapa para poder recibir estadísticas de dicha etapa.	Etapa 1, 2 y 3
BR-13	El jugador recibirá un máximo de 3 estrellas por cada nivel superado.	Etapa 1, 2 y 3
BR-14	El jugador recibirá un mínimo de 1 estrella por cada nivel superado.	Etapa 1, 2 y 3
BR-15	El jugador recibirá 1 estrella si pierde 2 intentos.	Etapa 1, 2 y 3
BR-16	El jugador recibirá 2 estrellas si pierde 1 intento.	Etapa 1, 2 y 3
BR-17	El jugador recibirá 3 estrellas si pierde 0 intentos.	Etapa 1, 2 y 3
BR-18	El jugador no recibirá estrellas cuando pierda sus 3 intentos.	Etapa 1, 2 y 3
BR-19	Las etapas se desbloquean acumulando cierta cantidad de estrellas (depende de la etapa).	Etapa 1, 2 y 3
BR-20	El jugador solo tiene 3 intentos (para equivocarse) por etapa.	Etapa 1, 2 y 3
BR-21	Si el jugador pierde los 3 intentos debe empezar nuevamente la etapa desde el nivel 1.	Etapa 1, 2 y 3
BR-22	El jugador solo puede jugar una etapa si ya ha sido desbloqueada.	Etapa 2 y 3
BR-23	El jugador solo puede jugar un nivel si ya ha sido desbloqueado.	Etapa 1, 2 y 3
BR-24	El jugador puede abandonar el nivel en el que se encuentre.	Etapa 1, 2 y 3
BR-25	El jugador puede reiniciar el nivel en el que se encuentre.	Etapa 1, 2 y 3

BR-2 6	Los jugadores pueden repetir las etapas y niveles para poder mejorar sus estadísticas cuantas veces quieran.	Etapa 1, 2 y 3
BR-2 7	Un jugador puede ver sólo sus propias estadísticas acerca de las etapas y niveles.	General
BR-2 8	El jugador que crea un grupo puede ver las estadísticas de los miembros de dicho grupo.	General
BR-2 9	Solo el jugador/usuario que cree el grupo puede eliminar los grupos que administra	General
BR-3 0	Solo el jugador/usuario que cree el grupo puede agregar nuevos usuarios del grupo que administra.	General
BR-3 1	Solo el jugador/usuario que cree el grupo puede eliminar los usuarios del grupo que administra.	General
BR-3 2	Las contraseñas serán mayores a 8 caracteres con signos de minúscula, mayúscula, numéricos.	General
BR-3 3	Formato de validación de correos.	General
BR-3 4	Alerta de inicio de sesión incorrecto	General

BR = Business rules.

## FM&A11 Ficha de juego

La ficha de juego confeccionada para el prototipo no funcional del juego serio SkillNet, contiene la siguiente información:

### ✓ Historia – Ayuda a cumplir el sueño de Robotín

Robotín es un dispositivo de red que siempre ha tenido un sueño, conocer la isla del Caribe (San Andrés - Colombia). Robotín es un equipo que se encuentra en un Centro de Datos en Limón (Costa Rica) y debe salir de su entorno para tomar una embarcación (WAN), eso sí, para salir no la tendrá tan fácil, debe superar una serie desafíos para poder pasar de su ubicación a la Isla.

El Centro de Datos está compuesto por salidas de seguridad, cada puerta podría abrirse en base a conocimientos en el área de Redes y Comunicaciones de

Datos. Para poder cumplir con el sueño de Robotín se tiene que pasar por 3 ambientes o escenarios, los cuales son los siguientes:

- Robotín está conectado en un MDF (*Main Distribution Frame*), para poder desconectarse y salir debe resolver desafíos sobre clases de direcciones del protocolo de internet, asignación de máscaras variables y subdivisión de redes, al completar estos ejercicios puede salir al salón principal de tecnologías de información.
- En el salón principal, la clave de salida es el ordenamiento e identificación del cableado necesario para la creación de una red y los dispositivos intermediarios que la componen, donde una vez que Robotín consiga ordenar dicho salón, puede salir y embarcarse en su travesía en el Crucero Island San Andrés.
- Ya en el crucero, debe Robotín enseñarle al capitán la trayectoria del viaje a San Andrés, para completar este desafío debe conocer la confección correcta de una red del protocolo de enrutamiento. Cada vez que supera un reto y desbloquea un nivel (paso de una ubicación a otra), la dificultad aumenta y recibe como premio un alimento básico de sobrevivencia (agua, proteína, medicamentos, entre otros).

✓ **¿Cómo se gana el juego?** – El jugador debe superar todas las actividades evaluativas de un nivel para lograr desbloquear el siguiente, en este caso se le abrirá una puerta que le permitirá seguir avanzando en el juego. Para ganar, debe desbloquear los 3 niveles y completar absolutamente todas las actividades en cada uno.

✓ **Estrategia de ludificación**

- **Sistema de puntuación** - Son asignados de acuerdo con el número de respuestas correctas al finalizar un desafío del juego, están basados en estrellas (Máximo 3 estrellas). Al completar un nivel el jugador podrá obtener un mínimo de 1 estrella si utilizó 2 intentos, 2 estrellas si utilizó 1 intento o 3 estrellas si no utilizó intentos.

- **Desafíos** - Resolución de preguntas tipo emparejar o asociar, entre otros.
- **Niveles** - División en tres niveles, de acuerdo con la dificultad de cada tipo de desafío: Nivel básico, intermedio y avanzado.
- **Intentos por etapa** - Cada vez que se inicie una etapa el jugador tendrá 3 intentos para todos los niveles de dicha etapa. Por cada respuesta incorrecta, el jugador perderá un intento. Si el jugador se queda sin intentos deberá volver a empezar desde cero la etapa en la que se encuentre y se le volverá a otorgar los 3 intentos. Lo anterior aplicará para cada etapa.
- **Sistema de ranking** - Se listará los jugadores de manera descendente o ascendente tomando como criterio las estrellas obtenidas por nivel + el tiempo de duración de la etapa. Se podrá filtrar, por nivel, también por etapa utilizando los mismos criterios, donde se sumarán las estrellas obtenidas entre los niveles que componen la etapa y de igual manera los tiempos.

### 9.2.5. Fase 5. Diseño de software – aspectos arquitectónicos

En relación con el diseño de software se realiza énfasis en diferentes aspectos característicos relacionados con la arquitectura.

#### FM&A12 Estilo arquitectónico

Entre los principales aspectos arquitectónicos característicos que se tomaron en cuenta en el diseño del prototipo del juego serio SkillNet están:

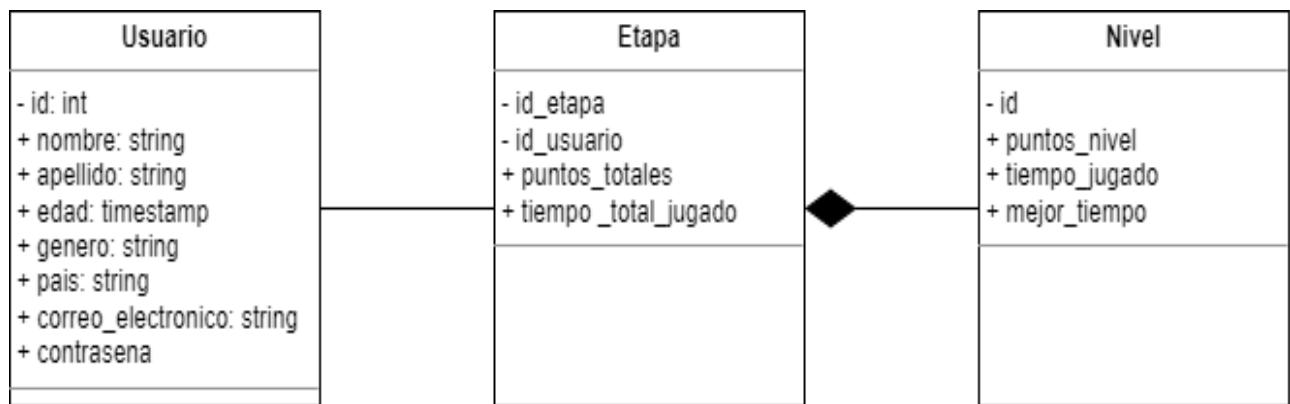
- ✓ **Modelo, vista, controlador** – En el prototipo no funcional de **SkillNet** se realiza una separación lógica por capas, donde en la capa modelo se desarrollan los objetos y se agrega la data, en la capa vista se agregan todas las interfaces de usuario y, finalmente, en la capa controlador se encarga de administrar las funcionalidades y el flujo de información entre las capas anteriores (modelo y vista).

- ✓ **Microservicios (web service)** – En el prototipo no funcional de SkillNet se consumirá recursos de la base de datos, tales como métricas e información del usuario, tiempo y puntuaciones.
- ✓ **Entorno de desarrollo** - El prototipo no funcional de **SkillNet** se desarrollará a través del IDE Visual Studio Code (VS Code) el cual es un editor refinado y optimizado para escribir y construir aplicaciones en casi cualquier tipo de lenguaje o Frameworks.
- ✓ **Framework o Librerías** - En esta ocasión se decide utilizar la librería de JavaScript conocida como React para el desarrollo del juego, ya que esta ofrece librerías que aportan funciones y que son de fácil utilidad para la interactividad del juego.
- ✓ **Gestor de base de datos** - Como gestor de base de datos para almacenar la información de los usuarios que se registran, sus puntuaciones y sus avances se hará a través de MySQL, ya que el juego no demanda mucha capacidad, este gestor es muy efectivo para el tipo de información que pretende almacenar.
- ✓ **Alojamiento** - Se decidió alojar la aplicación en Azure debido a que facilita las herramientas necesarias para un despliegue de la aplicación.
- ✓ **Autentificación de usuarios** - Para la autentificación de usuarios o logins se utilizará JWT (Json Web Token), una librería muy utilizada que cumple con los estándares de seguridad a la hora de permitir el acceso de un usuario.
- ✓ **Versionado** - Se mantiene una única versión del prototipo de la aplicación web en GitHub.

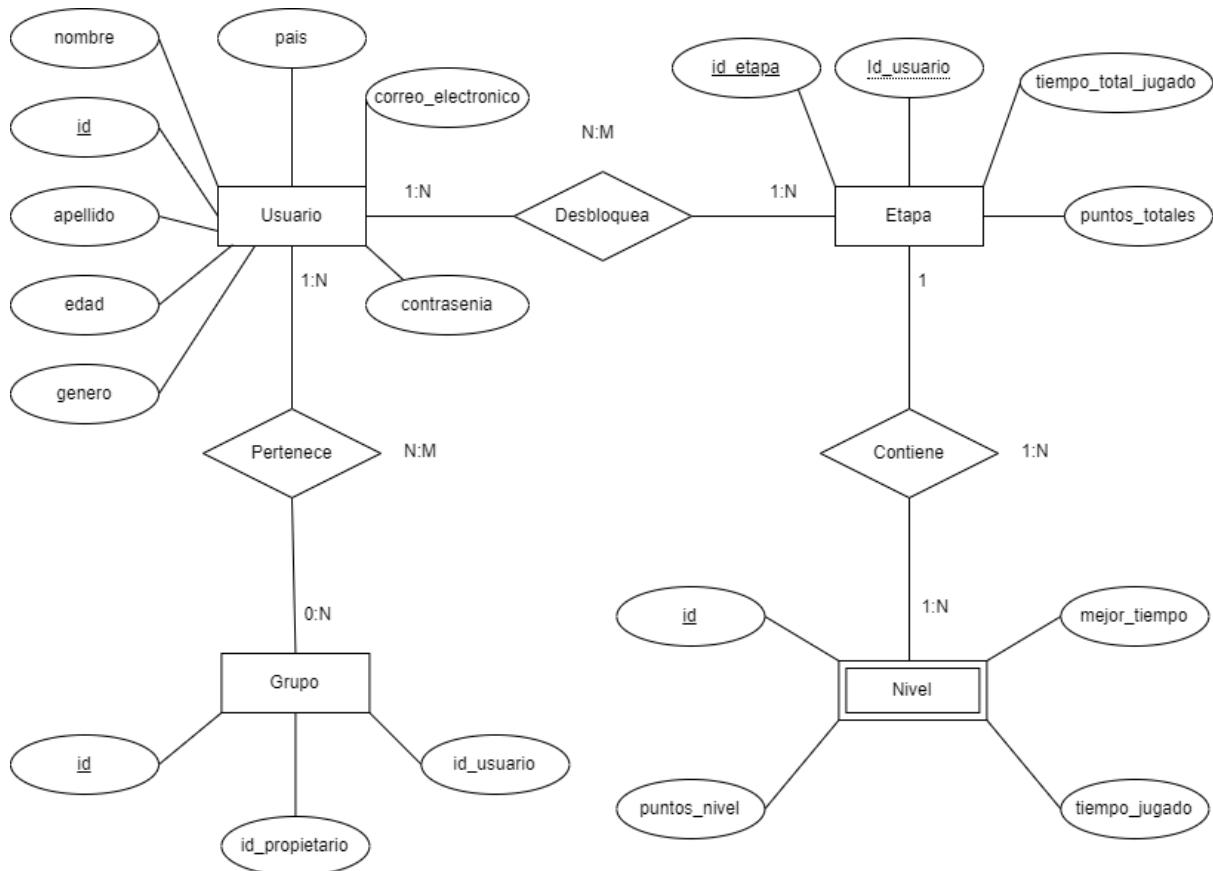
**FM&A13      Diagrama de clase (UML) y Diagrama Entidad Relación (ER)**

La Figura 9. 2 presenta a nivel gráfico el diagrama de clases o UML correspondiente al prototipo del juego serio SkillNet. Es importante indicar que el rombo en negro se refiere a una composición ya que, si la entidad etapa no existe, la entidad nivel tampoco.

**Figura 9. 2 Diagrama de clases**



La Figura 9. 3 muestra el diagrama entidad relación correspondiente al prototipo del juego serio SkillNet.

**Figura 9. 3 Diagrama entidad relación**

### 9.2.6. Fase 6. Codificación del software

Se debe estructurar el código mediante métodos o funciones específicas acorde con los requerimientos que se necesitan. Cuando un código es extenso y tiene operaciones que funcionan individualmente, se opta por modularizarlo para que sea sencillo de comprender y modificar, para un mejor mantenimiento. Como es una aplicación web se utiliza el lenguaje de etiquetas HTML junto a los estilos (Bootstrap, CSS, Bulma, entre otros) para estructurar las vistas “enmaquetado” de la página web, este junto con JavaScript permiten brindar dinamismo, crear eventos visuales, llamados a web services en la página.

#### FM&A14 Características del código de software

Al ser un código modularizado y simple, pensado para ser diseñado a través de HTML y CSS, permitiría aumentar el rendimiento del software.

## FM&A15 Reutilización de componentes

En el diseño del prototipo de SkillNet por ser una aplicación web contempla la reutilización de código y componentes dinámicos de software. Se da una reutilización a nivel componentes (Cards) a nivel de código (arrastre de las cards, entre otros componentes) y a nivel de objeto (botones, inputs, selects, entre otros).

### 9.2.7. Fase 7. Validación de la metodología

La validación es un aspecto de relevancia en el diseño y desarrollo de software, por ende, para el prototipo de juego no funcional de SkillNet, se propone una validación a través del juicio de expertos.

## FM&A16 Juicio de expertos

Para la respectiva validación de la metodología INTEGRA+506 se contó con profesionales expertos y con amplia experiencia en el área desarrollo web, diseño gráfico, expertos en el tema, expertos pedagógicos, entre otros, tal como se describe a continuación:

- ✓ Mag. Mainor Cruz Alvarado, analista, desarrollador web, UCR.
- ✓ M.Sc. Maynor Jiménez Castro, Matemático e Informático y docente, UCR.
- ✓ MBA. Rosa Julia Cerdas González, Contadora pública, docente, UCR.
- ✓ Lic. Olivier Blanco Sandí, Diseñador gráfico y desarrollador, UCR.
- ✓ Mag. Roberto Mesén Hidalgo, Enseñanza del Inglés y docente investigador, UCR.
- ✓ Lic. Itorino Espinoza Torres, Informático y docente investigador, UCR.
- ✓ Mag. Zoila Siles Navarro, experta en investigación y didáctica, Campus Sarapiquí, Universidad Nacional de Costa Rica.

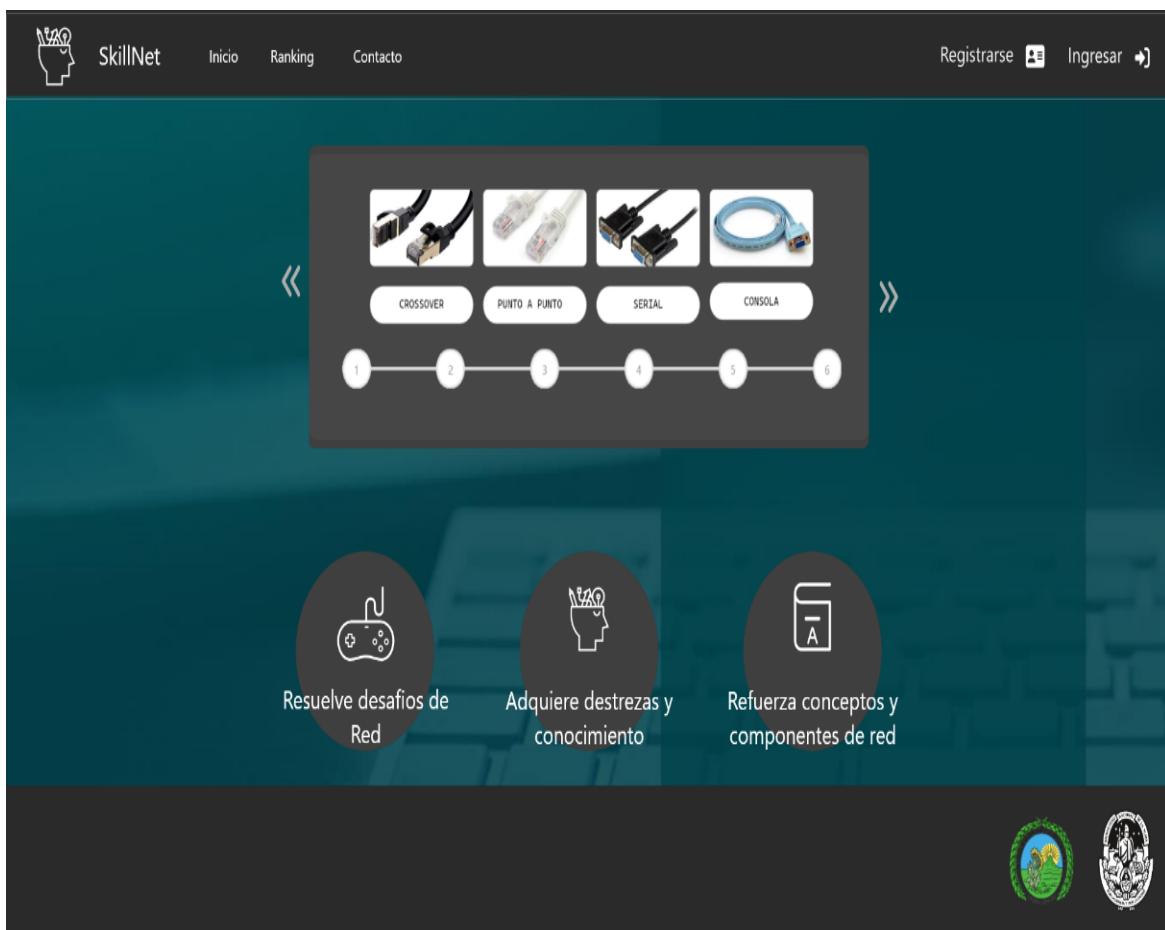
- ✓ Mag. Steven Cruz Sancho, analista y desarrollador web, Campus Sarapiquí, Universidad Nacional de Costa Rica.
- ✓ M.Sc. Cristian Brenes, Director del Recinto de Guápiles-UCR. Experto en análisis de sistemas.
- ✓ Mag. Carlos Felipe Escalante Solano, Experto en análisis y diseño de sistemas.

El panel de expertos validó a través de una encuesta la metodología planteada, así mismo, realizó observaciones que permitieron una mejora del juego.

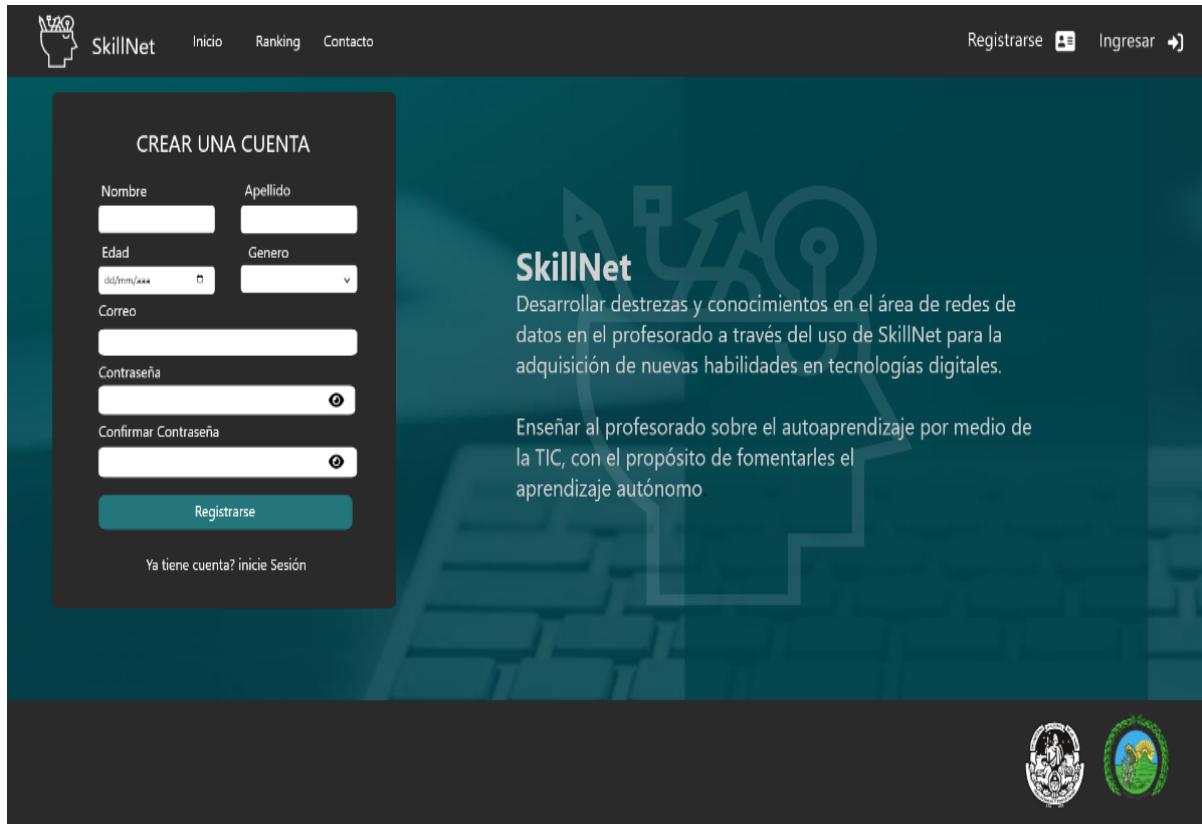
### 9.3. Interfaces del prototipo SkillNet

La Figura 9. 4 presenta la **interfaz principal** del prototipo de SkillNet la cual está conformada por la página de inicio, ranking, contactos, registrarse y el ingreso a la aplicación. En el botón de inicio se puede apreciar que la página contendrá un banner el cual ofrecerá una serie de imágenes relacionadas con el juego, así como brindar información en relación los desafíos, las destrezas, conocimientos por adquirir y los conceptos y componentes de red a reforzar a través de SkillNet.

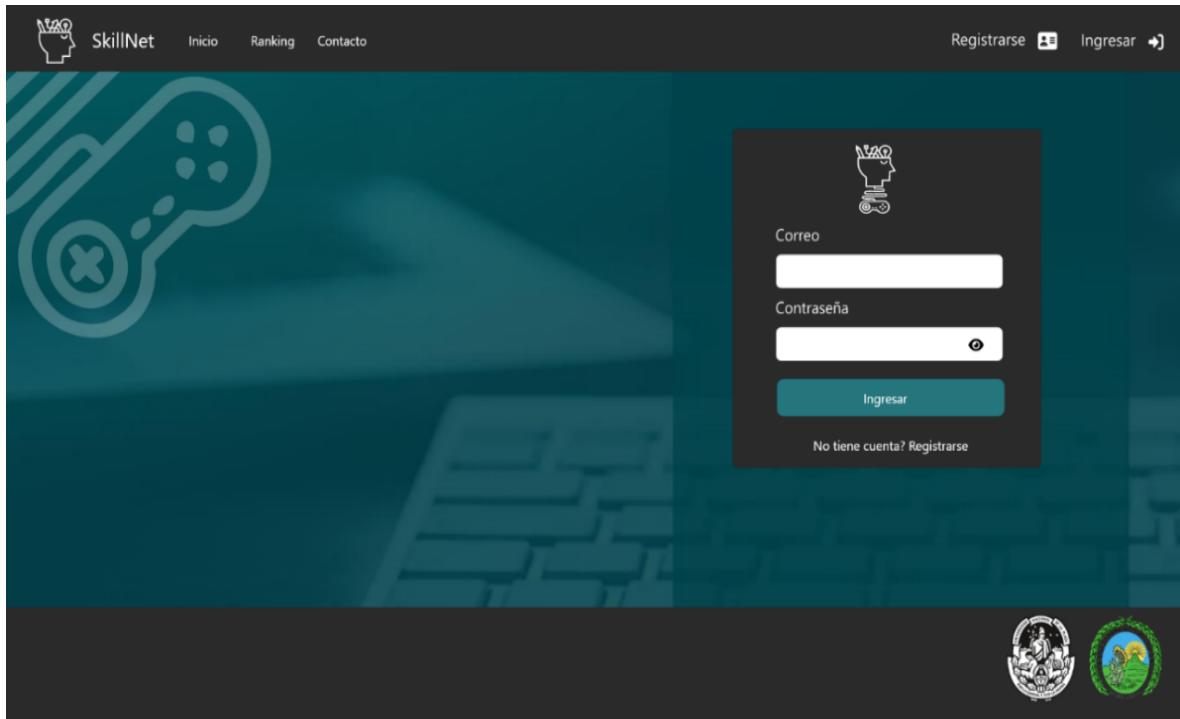
**Figura 9. 4** Página de inicio de SkillNet



La Figura 9. 5 presenta la interfaz de **registro** donde el usuario para poder acceder al juego serio debe crear una cuenta personalizada e individual, para ello debe llenar todos los campos solicitados en dicho formulario. Ver en la Tabla 9. 11 la Especificación de Requerimientos Funcionales del **Módulo Registro RF-1**.

**Figura 9. 5 Registro a SkillNet**

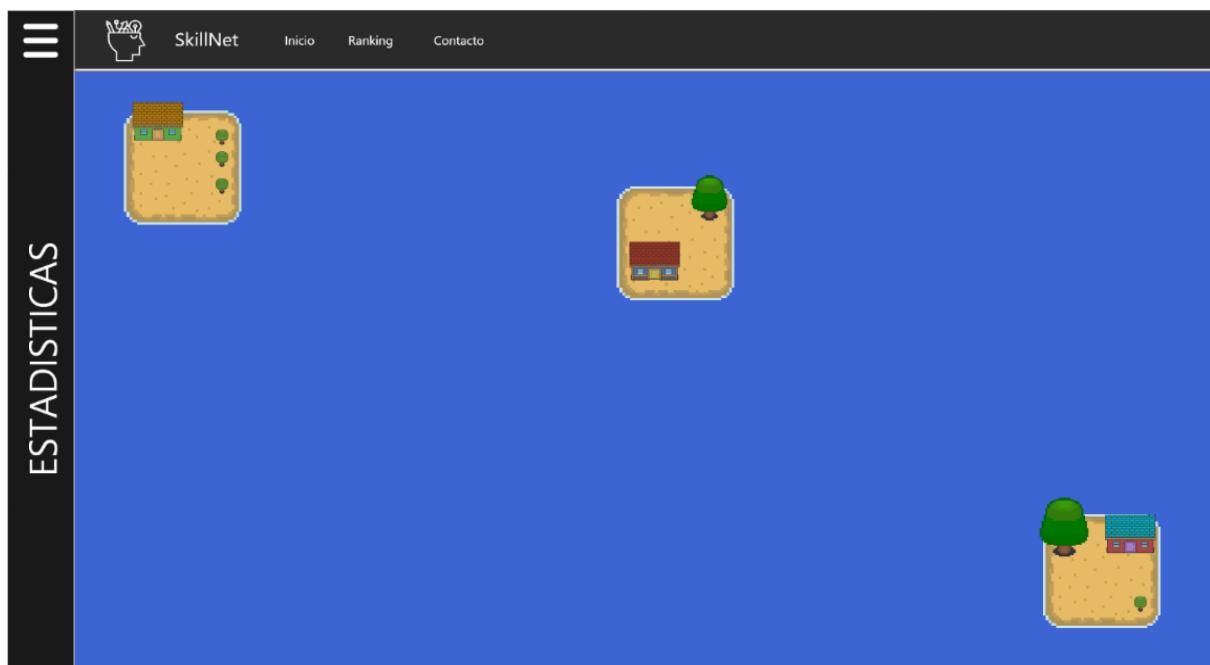
La Figura 9. 6 permite visualizar la interfaz de **ingreso a la aplicación**, el usuario previamente debe contar con un registro a la misma. Ver en la Tabla 9. 5 la Especificación de Requerimientos Funcionales del Módulo **Autentificación RF-2**.

**Figura 9. 6** *Ingreso a SkillNet*

La Figura 9. 7 presenta la pantalla general del juego, compuesto por 3 etapas de diferentes niveles, el jugador deberá ir desbloqueando progresivamente cada juego y etapa, iniciando desde el nivel 1, tal como se ha indicado en detalle en la Tabla 9. 8 la cual resume las Especificación de Requerimientos Funcionales del **Módulo Etapa 1 RF-5**.

Para no recargar el informe de imágenes, se presenta a modo de ejemplo únicamente 3 niveles de los 5 que componen la etapa 1, debido a que su funcionalidad es similar, aunque su intención pedagógica sea distinta.

Figura 9. 7 Pantalla principal del juego en general



La Figura 9. 8 presenta a modo de ejemplo las estadísticas en relación con cada uno de los jugadores, donde se toma en cuenta el nivel de avance por competencia, insignias, puntaje obtenido, el tiempo tomado para la resolución de cada desafío, así como resultados a nivel individual, según nacionalidad, edad y género. Igualmente, se puede apreciar que el juego se puede realizar de forma individual o grupal garantizando una separación lógica de los datos por jugador.

Figura 9. 8 Estadísticas a nivel individual en SkillNet

**ESTADISTICAS**

CARLOS

Puntos ganados: 20  
Tiempo Jugado: 00:00:00  
Mejor Tiempo: 00:00:00

**CREAR GRUPO**

#	Nombre	Correo Electronico	Actions
1	Steven R.	ejemplocorreo@dominio.com	
2	Isaac C.	ejemplocorreo@dominio.com	
3	Juan T.	ejemplocorreo@dominio.com	
4	Patricia B.	ejemplocorreo@dominio.com	
5	Juan Carlos S.	ejemplocorreo@dominio.com	
6	Steven S.	ejemplocorreo@dominio.com	
7	Minor	ejemplocorreo@dominio.com	

En la Figura 9. 9 se puede observar a modo de ejemplo las opciones para realizar las **estadísticas a nivel grupal**, donde el usuario crea y administra grupos (ver detalles, observar estadísticas o eliminar un grupo si así lo desea). En relación con las estadísticas grupales, se podría analizar el rango etario, género, puntaje obtenido, nacionalidades y seguimiento de errores, entre otros, tal como se puede

observar en la Figura 9. 10, en la cual se muestran datos de los jugadores en relación con el rango etario, nacionalidad y género.

**Figura 9. 9 Administración de grupos en SkillNet**

The screenshot shows a dark-themed web interface for managing groups. At the top, there's a navigation bar with a logo, the text "SkillNet", and links for "Inicio", "Ranking", and "Contacto". Below this is a search bar labeled "Nombre del grupo" and a teal "Agregar" button. The main content area is a table with the following data:

#	Nombre	Acciones
1	Clase A-1	
2	Clase B-3	
3	Taller Redes 01	
4	Clase B-2	
5	Clase A-2	
6	Taller Redes 02	
7	Clase xxxx	

**Figura 9. 10 Estadísticas a nivel grupal**

En la Figura 9. 11 se puede observar una ventana de información teórica relacionada con el contenido de la habilidad o competencia por potenciar con el respectivo juego de la etapa que va a realizar. Es decir, cuando el jugador empiece la etapa 1 - nivel 1, el sistema mostrará dos pantallas con información acerca de clases de red A, B y C para que el jugador adquiera conocimientos asociados al nivel actual y pueda aplicarlos en el juego. La primera pantalla con información tiene un botón “siguiente” que el jugador puede presionar para que el sistema muestre la siguiente pantalla con más información sobre las clases de red.

La segunda pantalla Figura 9. 12 presenta más información sobre las clases de red, tiene un botón “siguiente” que el jugador puede presionar para que el sistema muestre la siguiente pantalla la cual contiene las instrucciones para jugar el nivel actual. Esta pantalla con instrucciones acerca de cómo mover los recuadros también tendrá un botón “siguiente” el cual llevará al jugador a empezar a jugar el nivel 1.

**Figura 9. 11** Información teórica del contenido de la etapa 1 nivel 1

**Información para el nivel 1**

**Objetivo:** Potenciar conocimientos básicos en la identificación de clases de red.

**Clase A:** 8 – 20 Rango máscaras. R. H. H. H  
Esta clase de red se utiliza comúnmente en redes extensas, grandes. Del IP con un primer octeto a partir de 1 al 126 son parte de esta clase.  
Los otros tres octetos son usados para identificar cada host.

**Clase B:** 16 – 28 Rango máscaras. R. R. H. H  
– Se utiliza para las redes de tamaño mediano. Un buen ejemplo es un campus grande de la universidad. Las direcciones del IP con un primer octeto a partir del 128 al 191 son parte de esta clase. Las direcciones de la clase B también incluyen el segundo octeto como parte del identificador neto.

**Clase C:** 24 – 32 Rango máscaras. R. R. R. H  
Las direcciones del IP con un primer octeto a partir del 192 al 223 son parte de esta clase.  
Las direcciones de la clase C, también incluyen a segundos y terceros octetos como parte del identificador neto. Utilizan al último octeto para identificar cada host.

**Siguiente**

Igualmente, en la [Figura 9.12](#) se puede observar a modo de ilustración uno de los ejemplos prácticos que ofrece el juego al usuario con el propósito de empoderar el contenido teórico de etapa 1 – nivel 1.

**Figura 9. 12** Ventana informativa ejemplos prácticos de la etapa 1 – nivel 1

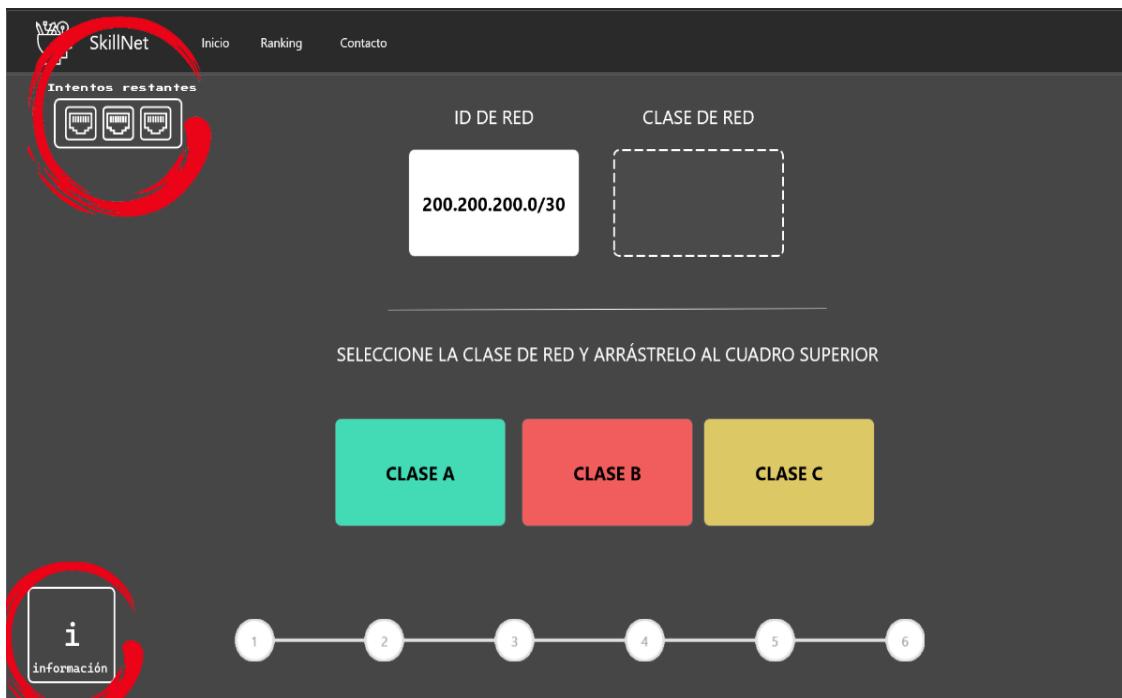
**Información para el nivel 1**

Clase	Intervalo	N. de Redes	N. de hosts por red	Máscara de red	Id Broadcast
A	8.0.0.0 – 127.255.255.255	128	16777214	255.0.0.0	x.255.255.255
B	128.0.0.0 – 191.255.255.255	16384	65534	255.255.0.0	x.x.255.255
C	192.0.0.0 – 255.255.255.255	2097152	254	255.255.255.0	x.x.x.255

**Siguiente**

La Figura 9. 13 muestra a modo de ejemplo el ejercicio práctico de la etapa 1 – nivel 1 el cual muestra 5 recuadros, en el primero, se muestra un id de red y en el segundo se arrastra una de las 3 opciones (Clase A, Clase B o Clase C). Cuando el jugador acierte la opción correcta, el sistema generará un nuevo id de red y se repite el proceso hasta completar los 6 pasos de dicho nivel. En la misma pantalla al igual que las siguientes se puede apreciar que el jugador cuenta con 3 vidas y una ventana de información (i) que permite repasar el contenido teórico de la Figura 9.12.

**Figura 9. 13 Ejercicio práctico de la etapa 1 - Nivel 1**



En la Figura 9. 14 se puede apreciar un ejemplo de los mensajes informativos que se cerrarán automáticamente después de algunos segundos. Cuando el jugador acierte, el sistema muestra un mensaje de felicitaciones para motivarle a continuar aprendiendo.

**Figura 9. 14 Mensaje de felicitaciones por completar correctamente el nivel 1**

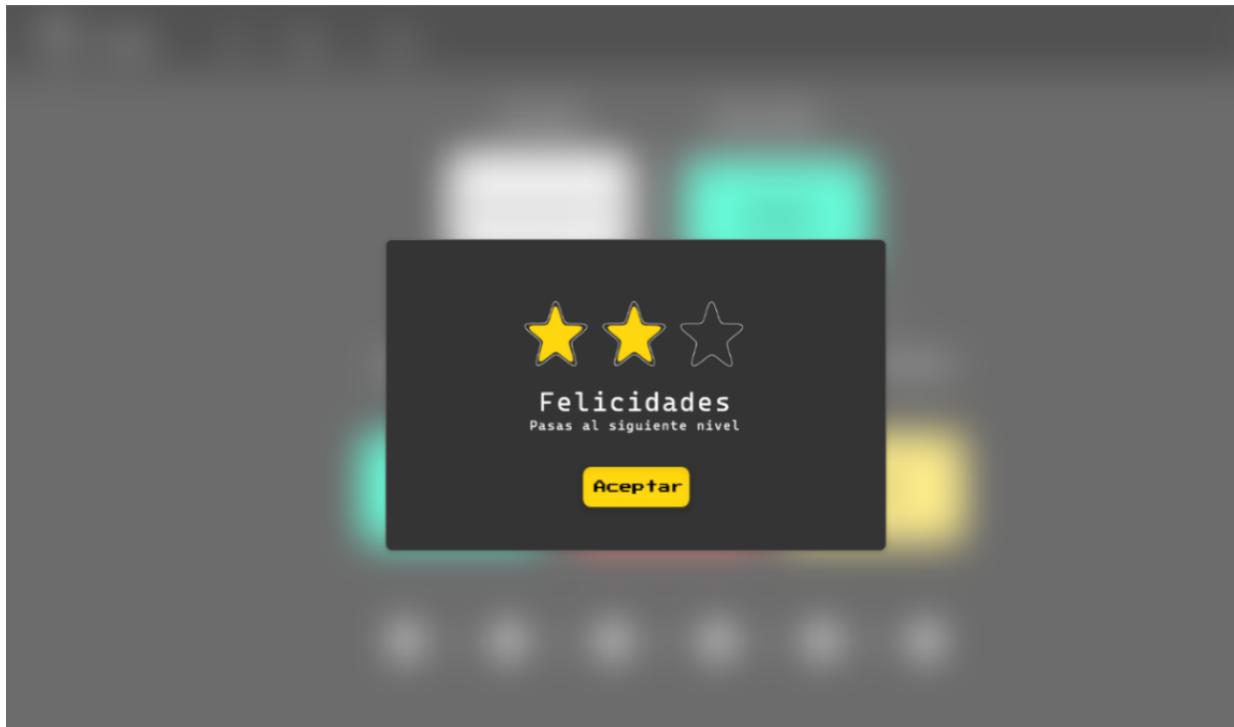
En la Figura 9. 15 se puede apreciar un ejemplo de los mensajes informativos que se despliegan cuando el jugador se equivoca o desacierta, el sistema muestra un mensaje de respuesta incorrecta con una pista o información para que el jugador analice cuál podría ser la respuesta correcta.

**Figura 9. 15 Mensaje informativo de desacuerdo en el nivel 1 – etapa 1**

La Figura 9. 16, muestra el cuadro que se les desplegará al jugador al completar el desafío de un nivel, igualmente, la cantidad de estrellas obtenidas y un

botón “aceptar”, el cual el jugador debe presionar para continuar al próximo reto, así sucesivamente hasta completar cada etapa con todos sus niveles.

**Figura 9. 16** Mensajes informativo de avance de nivel y estrellas ganadas



La Figura 9. 17 cuando el jugador inicia el nivel 2, el sistema mostrará una pantalla con información acerca de máscaras por defecto para que el jugador adquiera conocimientos asociados al nivel actual y pueda aplicarlos en el juego. La primera pantalla con información tendrá un botón “siguiente” el cual llevará al usuario a iniciar el juego según nivel actual.

Figura 9. 17 Información teórica y práctica del contenido de la etapa 1 - nivel 2

**MASCARA POR DEFECTO**

Objetivo: Reforzar conocimientos vinculados con la identificación de diferentes tipos de máscaras según la clase de red.

Las redes de **"Clase A"** utilizan una máscara de subred predeterminada de **255.0.0.0** y tienen de 0 a 127 como su primer octeto. Por ejemplo, la dirección 10.52.36.11 es una dirección de clase A. Su primer octeto es 10, que está entre 1 y 126, ambos incluidos.

Las redes de **"Clase B"** utilizan una máscara de subred predeterminada de **255.255.0.0** y tienen de 128 a 191 como su primer octeto. Por ejemplo, la dirección 172.16.52.63 es una dirección de clase B. Su primer octeto es 172, que está entre 128 y 191, ambos inclusive.

Las redes de **"Clase C"** utilizan una máscara de subred predeterminada de **255.255.255.0** y tienen de 192 a 223 como su primer octeto. Por ejemplo, la dirección 192.168.123.132 es una dirección de clase C. Su primer octeto es 192, que está entre 192 y 223, ambos incluidos.

**Siguiente**

**Intentos restantes**

ID DE RED	MASCARA
192.193.194.0/29	CLASE C 255.255.255.0

✓ Respuesta correcta!  
¡Sigue así, vas muy bien!

Aceptar

SELECCIONE LA MÁSCARA DE RED Y ARRÁSTRELA AL CUADRO SUPERIOR

**CLASE A**  
255.0.0.0

**CLASE B**  
255.255.0.0

**i**  
información

1 — 2 — 3 — 4 — 5 — 6

La Figura 9. 18 muestra a modo de ejemplo práctico el ejercicio que ha sido resuelto satisfactoriamente para el tema de broadcast de red, igualmente, se aprecia el mensaje de refuerzo positivo por lograr superar el desafío del nivel.

**Figura 9. 18** Información teórica y práctica del contenido de la etapa 1 - nivel 3

The screenshot shows a game interface for 'SkillNet' with a red header bar containing the chapter title. Below it, there are two main sections: a theoretical explanation and a practical exercise.

**Theoretical Section (Broadcast):**

- BROADCAST:** A heading in bold capital letters.
- Text:** "La broadcast IP permite enviar paquetes de datos a través de una red de ordenadores a todos los usuarios de una red local. En cuanto a las direcciones individuales de cada uno, no son necesarias para realizar la transmisión de datos. En caso de que sea necesario, la dirección broadcast se puede calcular fácilmente."
- Text:** "Para obtener el broadcast solamente tenemos que tomar el id de red y sumarle el salto de red, por lo tanto debemos saber como obtener el salto de red."
- Text:** "Ejemplo:  
Id de red: 220.221.222.0/30  
Salto: se obtiene elevando el "2" a la cantidad de bits que se encuentran del lado derecho del slash y en el mismo octeto. En este caso, el id de red tiene un slash de 30 por lo que simplemente tenemos 2 bits hacia la derecha del slash y en el mismo octeto y al resultado le restamos 1, por lo tanto  $2^2 = 4$  y  $4-1 = 3$ . Entonces Salto = 3.  
Por lo tanto el Broadcast es: 220.221.222.3/30"

A yellow button labeled "Siguiente" (Next) is located at the bottom of this section.

**Practical Exercise Section:**

- Intentos restantes:** Shows three icons representing attempts remaining.
- ID DE RED:** A white box containing the text "190.191.192.0/29".
- BROADCAST:** A green box containing the text "190.191.192.7".
- Feedback:** A message box says "Respuesta correcta! ¡Sigue así, vas muy bien!" with a green checkmark icon, and a blue "Aceptar" (Accept) button.
- Text:** "SELECCIONE LA MASCARA Y ARRÁSTRELO AL CUADRO SUPERIOR"
- Options:** Two colored boxes for dragging: a red one with "190.191.192.3" and a yellow one with "190.191.192.15".
- Information:** A small box with an 'i' icon labeled "información".
- Progress:** A horizontal progress bar with six circles numbered 1 to 6.

En la Figura 9. 19 se pue apreciar la ventana de información que se le despliega al jugador en relación con el objetivo y los contenidos teóricos de la fase por superar en la etapa 2 - nivel 1. La etapa 2 cuenta con 2 niveles.

**Figura 9. 19** *Información teórica etapa 2 - nivel 1*



En la Figura 9. 20 y Figura 9. 21 se pueden observar las actividades de la etapa 2 - nivel 1 donde el sistema mostrará 4 recuadrosopcionales, donde se debe evaluar según los dispositivos mostrados, el tipo de cableado idóneo para interconectarlos correctamente, donde la opción a elegir se debe arrastrar al recuadro para ser validada. Luego, el sistema generará una nueva opción de dispositivo de red y se repite el proceso hasta completar el total de los pasos (6) de dicho nivel.

Figura 9. 20 Actividades etapa 2 - nivel 1 ejemplo A

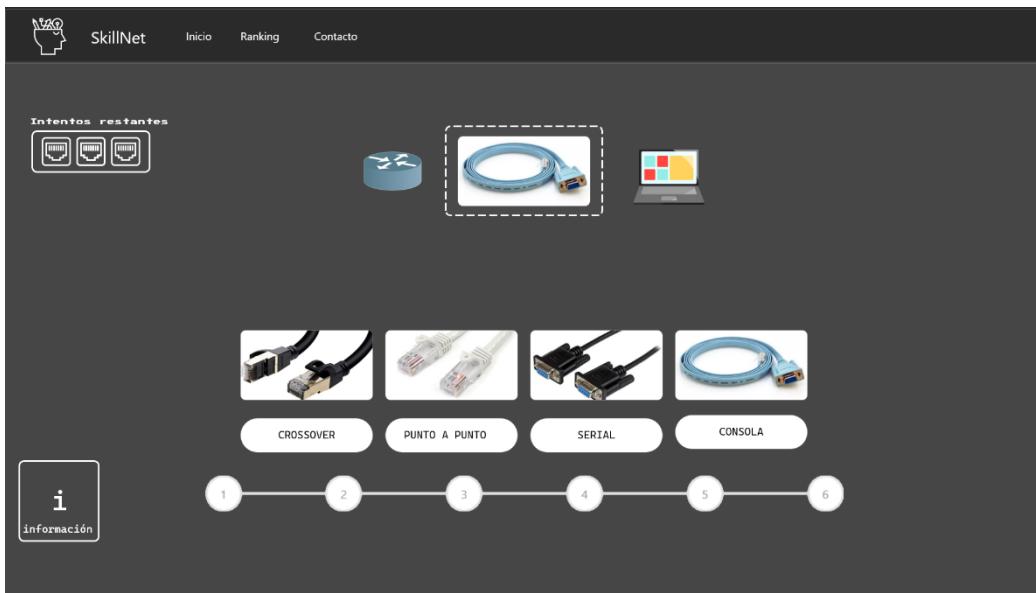
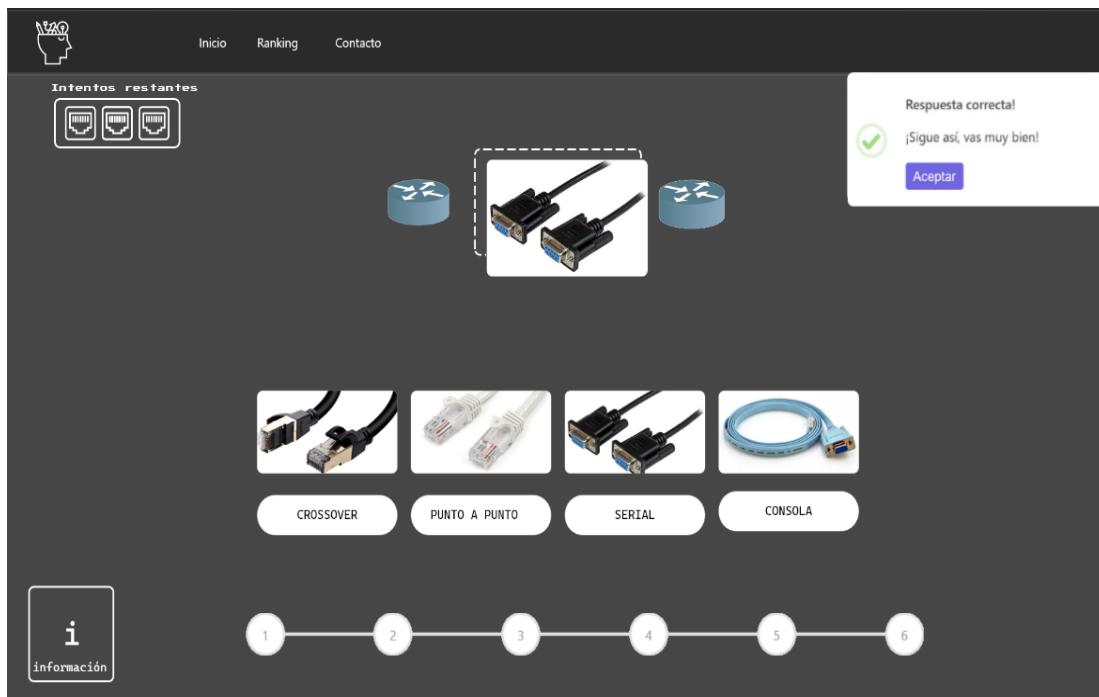
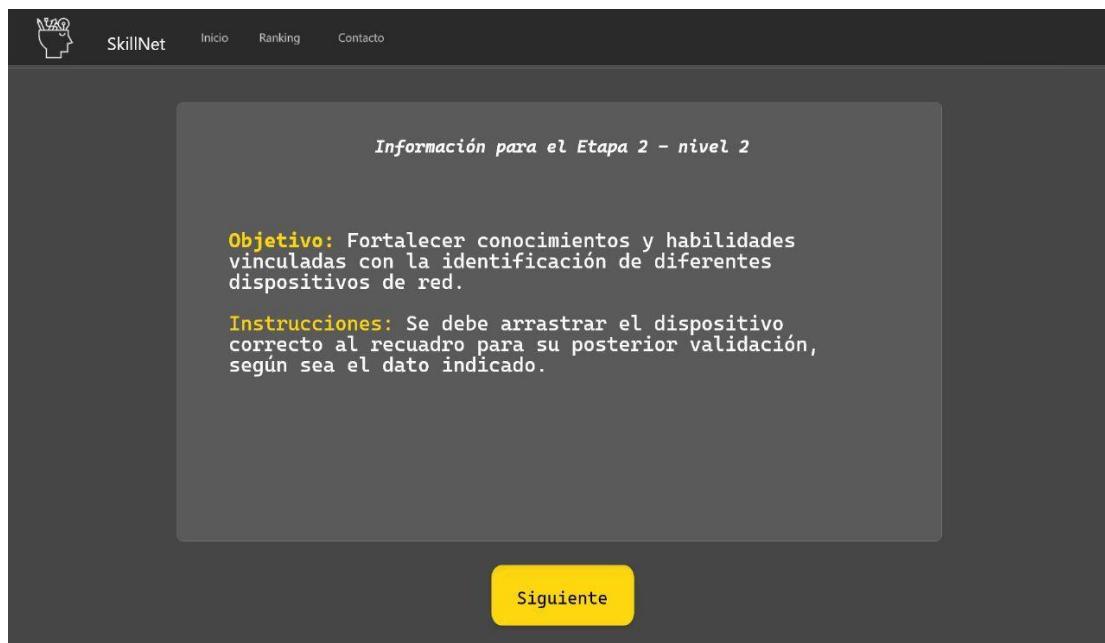


Figura 9. 21 Actividades etapa 2 - nivel 1 ejemplo B



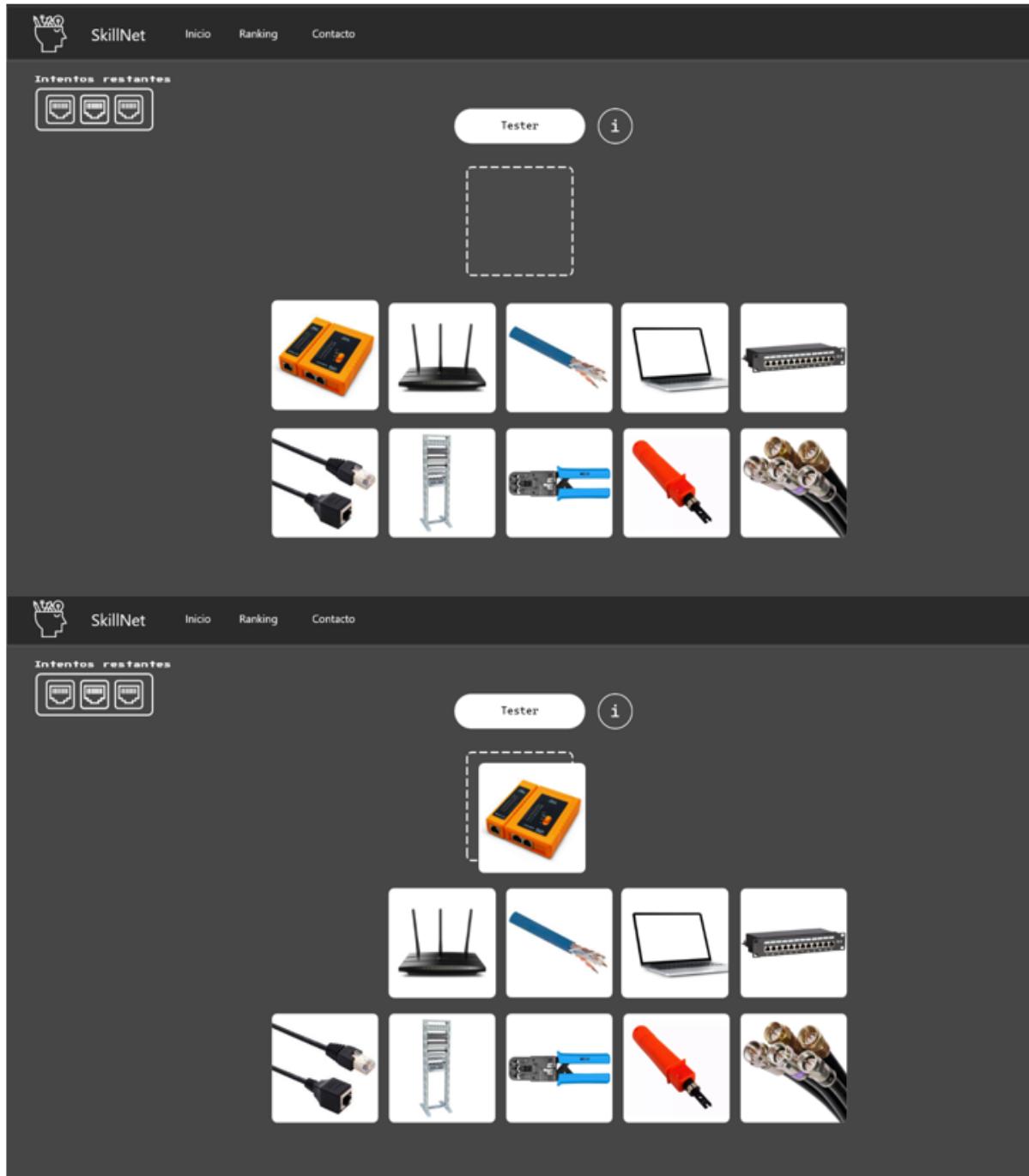
En la Figura 9. 22 se pue apreciar la ventana de información que se le despliega al jugador en relación con el objetivo y los contenidos teóricos de la fase por superar en la etapa 2 - nivel 2.

**Figura 9. 22** *Información etapa 2 - nivel 2*



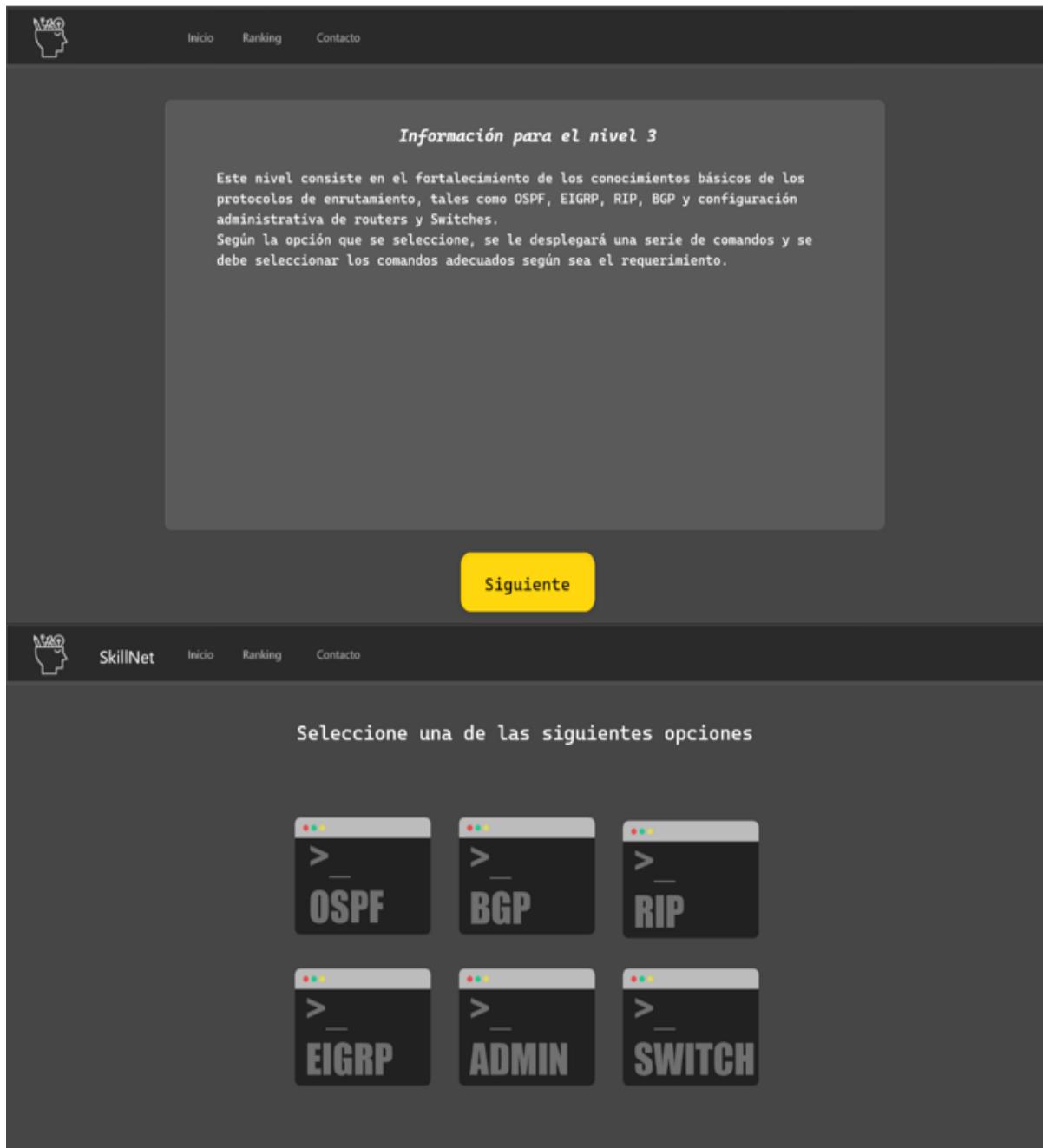
La Figura 9. 23 muestra a modo de ejemplo los ejercicios prácticos de la etapa 2 – nivel 2, en la cual el jugador debe arrastrar los diferentes dispositivos a la casilla que contiene el nombre, acorde con la información solicitada. Una vez superado este desafío, debe realizar la misma acción por cada uno de los dispositivos.

Figura 9. 23 Ejercicios prácticos etapa 2 - nivel 2



En Figura 9. 24 se pude observar de forma gráfica la información básica relacionada algunos de los protocolos de enrutamiento. Para superar este nivel, el usuario debe seleccionar primeramente un equipo o protocolo de red para realizar la respectiva configuración, puede elegir entre los protocolos RIP, EIGRP, OSPF o BGP, además, elegir entre la configuración administrativa o bien, un switch.

Figura 9. 24 *Información general para la etapa 3*



La Figura 9. 25 permite observar la información de la etapa 3 – nivel 1, la cual consta de un único nivel, donde se debe seleccionar de una serie de comandos relacionados con la configuración de diferentes protocolos de comunicación, los que permitan brindar respuesta a cada interrogante planteada según el protocolo o equipo elegido por el usuario.

**Figura 9. 25** Información para la etapa 3 - nivel 1

The screenshot shows a user interface for the SkillNet game. At the top, there's a red header bar with the text 'CAPÍTULO IX. PROTOTIPO NO FUNCIONAL DEL JUEGO SERIO SKILLNET'. Below this, the main page has a dark background with white text. A navigation bar at the top includes a logo, the text 'SkillNet', and links for 'Inicio', 'Ranking', and 'Contacto'. The main content area contains a large gray box with the heading 'Información para el Etapa 3 - nivel 1'. Inside this box, there are two sections: 'Objetivo' and 'Instrucciones'. The 'Objetivo' section states: 'Objetivo: Fortalecer conocimientos y habilidades vinculadas con la identificación de configuraciones de red.' The 'Instrucciones' section states: 'Instrucciones: Se debe seleccionar de una serie de comandos relacionados con la configuración de diferentes protocolos de comunicación, los que permitan brindar respuesta a cada interrogante planteada.' Below this gray box is a yellow button labeled 'Siguiente'. At the bottom of the page, there's another navigation bar with the same structure as the top one.

**Objetivo:** Fortalecer conocimientos y habilidades vinculadas con la identificación de configuraciones de red.

**Instrucciones:** Se debe seleccionar de una serie de comandos relacionados con la configuración de diferentes protocolos de comunicación, los que permitan brindar respuesta a cada interrogante planteada.

Siguiente

El/los comandos adecuados para la actualización de paquetes en el protocolo OSPF son: (elegir 2 de la lista).

**OSPF**

```

> - ip ospf hello-interval 5
> - ip ospf dead-interval 20

```

- router bgp 300
- network 205.205.205.32 mask 255.255.255.224
- interface loopback 50
- ip add 50.50.50.51 255.255.255.255
- router rip
- ip ospf hello-interval 5
- ip ospf dead-interval 20
- version 2
- ip ospf message-digest-key 1 md5 7 asecret
- no shutdown

En la Figura 9. 26 se puede apreciar el sistema de registro utilizado para el ranking implementado en el prototipo de SkillNet, el cual permite visualizar los primeros 100 jugadores de manera descendente o ascendente tomando como criterio las estrellas obtenidas por nivel más el tiempo de duración para resolver cada etapa. Además, permite realizar filtros por nivel y por etapa haciendo uso de

los mismos criterios, pero para este filtro se sumarán las estrellas obtenidas entre los niveles que componen la etapa y de igual manera los tiempos.

Figura 9. 26 Sistema de Ranking

The screenshot shows the 'Ranking' section of the SkillNet game. At the top, there is a navigation bar with the SkillNet logo, 'SkillNet', 'Inicio', 'Ranking', 'Contacto', 'Registrarse', and 'Ingresar'. Below the navigation bar, there is a decorative graphic featuring three stylized human figures in yellow, green, and red, each with the number '2', '1', and '3' respectively. To the right of the graphic is a table displaying player statistics:

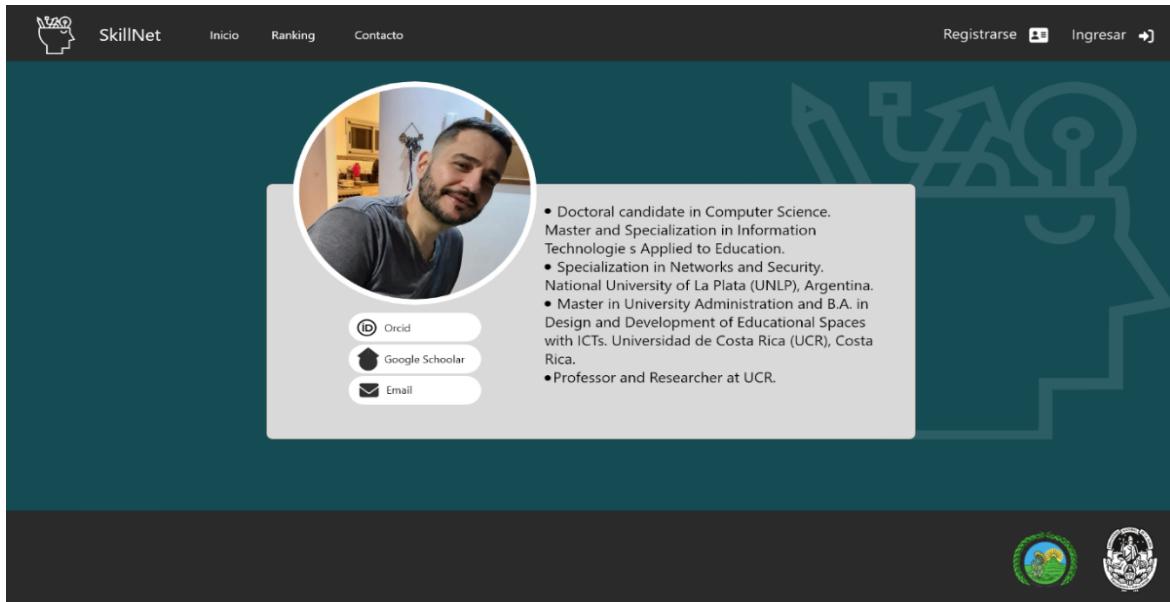
Nombre	País	Puntos	Tiempo
Juan	Costa Rica	100	00:00:00
Marta	Argentina	100	00:00:00
Michel	Colombia	92	00:00:00
Carlos	Argentina	90	00:00:00
Miguel	Perú	90	00:00:00
Samuel	Costa Rica	90	00:00:00
Nidia	Nicaragua	87	00:00:00

• • •

In the bottom right corner, there are two small icons: the Costa Rican flag and the coat of arms of Costa Rica.

En la Figura 9. 27 se presenta la interfaz de información de contacto, la cual le permitirá al jugador realizar consultas en relación con el juego, solicitar sus métricas o estadísticas de avance, así como resolver dudas generales, además, se le facilita el acceso a información del crear del juego, tal como resultados de las investigaciones realizadas (publicaciones) así como las áreas de interés.

Figura 9. 27 *Información de contacto*



En resumen, se han presentado algunas de las interfaces del prototipo del juego serio SkillNet, el cual consta de 3 fases y cada fase de 5 niveles con sus distintos desafíos que permiten adquirir las habilidades y conocimientos intermedios en redes de datos.

#### 9.4. Conclusiones del capítulo

Se diseña un prototipo no funcional del juego serio denominado **SkillNet**, el cual cumple con las siguientes características funcionales:

- ✓ **Tipo de aplicación** - Es una aplicación de tipo web con acceso desde cualquier browser con conexión a internet y altamente disponible.
- ✓ **Arquitectura** – Es una arquitectura tipo *multi-tenant*, es decir, ofrece una separación lógica de datos de cada usuario.
- ✓ **Colaboración** - Es una aplicación altamente colaborativa entre usuarios.
- ✓ **Disponibilidad** - Es una aplicación altamente disponible desde cualquier lugar y momento, sin importar el espacio físico.
- ✓ **Versionado** – Es una aplicación de versión simple y compartida para todos los clientes.
- ✓ **Personalización** – Es una aplicación que ofrece la opción de que cada *tenant* personalice y configure su perfil de usuario.
- ✓ **Interactividad** – Es una aplicación que considera ofrecer interactividad totalmente sincrónica.
- ✓ **Seguridad y acceso** – Es una aplicación que ofrece seguridad y control de acceso por usuario.
- ✓ **Suscripción** – Es una aplicación que ofrece un manejo de cuentas de usuario, *loggings*, control de uso y métricas.
- ✓ **Ubicuidad** – Es una aplicación que ofrece servicios y recursos multimediales en tiempo real.

- ✓ **Versionado** - Es una aplicación que ofrece una versión simple, es decir, es una única versión compartida para todos los clientes.
- ✓ **Guía o ayuda** - Es una aplicación que ofrece video tutoriales en diferentes en español.

A nivel general, la metodología INTEGRA+506 cumple con el objetivo de guiar el diseño de un prototipo no funcional de un juego serio orientado a la formación de competencias tecnológicas en el usuario final, en este caso SkillNet.

# CAPÍTULO X. CONCLUSIONES Y LÍNEA DE TRABAJOS FUTUROS

## 10.1. Introducción

Este capítulo reúne las principales conclusiones generales de todo el trabajo, específicamente, acerca de la propuesta metodológica y arquitectónica para el diseño de juegos serios orientados a la formación de competencias tecnológicas. Por último, se detallan las líneas de trabajo o de investigación futuras que servirán para potenciar en el profesorado los conocimientos generados en esta investigación.

## 10.2. Conclusiones de la tesis

El objetivo general de esta investigación consistió en definir una metodología y arquitectura de software para el diseño de juegos serios que permitan la adquisición o formación de competencias tecnológicas. Luego, a partir de la metodología propuesta denominada INTEGRA+506 se diseñó un prototipo no funcional de un juego serio denominada SkillNet, el cual se enfocó en potenciar competencias digitales en el profesorado, específicamente las vinculadas con conocimientos básicos en redes de datos. Con este propósito, se plantearon una serie de objetivos específicos abordados en detalle en el capítulo 1, a saber:

1. Estudiar los conceptos básicos vinculados con las TIC, competencias tecnológicas, juegos serios y desarrollo de software.
2. Describir antecedentes de metodologías utilizadas en el diseño de software, en particular de juegos serios, así como las arquitecturas de software y de despliegue utilizadas.
3. Elaborar una propuesta metodológica y arquitectónica para el diseño de juegos serios que potencien la adquisición de competencias tecnológicas.
4. Validar la metodología a través de un panel de expertos o a través del diseño de un prototipo no funcional de juego serio, de manera que permita analizar el alcance del aporte de esta tesis y las posibles mejoras por realizar.
5. Elaborar un informe con los resultados y las conclusiones a las que se arriba.

Los objetivos específicos 1 y 2, se abordaron a través de una exhaustiva revisión sistemática de literatura, la cual permitió brindar conclusiones teóricas, mismas que se describen y analizan en detalle en los Capítulos 2, 3, 4, 5, 6 y 7. Igualmente, el objetivo específico 3 se abordó en el capítulo 8, a través del cual se

definió una propuesta metodológica y arquitectónica para el diseño de un juego serio para potenciar la adquisición de competencias tecnológicas. Luego, el objetivo 4 se desarrolló en el capítulo 9 a través del diseño de un prototipo no funcional de un juego serio llamado SkillNet, el cual está orientado a la formación de competencias digitales, específicamente las relacionadas con conceptos básicos en redes de datos. Finalmente, el objetivo 5 correspondiente al informe de resultados, se describe en el capítulo 9. En esta línea, se retoman y describen en detalle las trece (13) preguntas de investigación (PI) que se definieron en el Capítulo 1, así como las conclusiones relacionadas a estas preguntas, mismas según trabajo de investigación realizado.

- Plx1 ¿Cómo se define o describe el concepto de Tecnologías de la Información y la Comunicación (TIC)?
- Plx2 ¿Cómo se define el concepto de competencias o competencias tecnológicas?
- Plx3 ¿Cómo se definen actualmente los juegos serios?
- Plx4 ¿Cuáles son las principales características, requerimientos técnicos y/o funcionales de los juegos serios?
- Plx5 ¿Qué antecedentes existen de la utilización de juegos serios para la generación y desarrollo de competencias tecnológicas?
- Plx6 ¿Cómo se define o describe el concepto de *cloud computing*?
- Plx7 ¿Cuáles son las características esenciales de *cloud computing*?
- Plx8 ¿Cómo se describen los modelos de servicio y despliegue en *cloud computing*?
- Plx9 ¿Qué aspectos se deben considerar en el desarrollo de software?, ¿Cómo se describen o caracterizan estos aspectos?
- Plx10 ¿Cuáles metodologías se han utilizado para el diseño de juegos serios?, ¿qué antecedentes existen en este sentido?
- Plx11 ¿Qué aspectos metodológicos se consideran en la literatura para el diseño de juegos serios en la formación de competencias tecnológicas?
- Plx12 ¿Cuáles arquitecturas se han utilizado para el diseño de videojuegos o juegos serios?, ¿qué antecedentes existen en este sentido?

Plx13 ¿Qué aspectos arquitectónicos se consideran en la literatura para el diseño de videojuegos o juegos serios?

En relación con las preguntas de investigación 1 y 2 vinculadas el tema de las TIC y competencias tecnológicas, se definieron y caracterizan ampliamente en el capítulo II, lo cual permitió identificar su importancia e impacto en la formación de competencias tecnológicas, así como, a partir de sus aspectos característicos brindar una definición propia de las mismas.

Las interrogantes 3, 4 y 5 relacionadas con los juegos serios, se abordaron en detalle en el capítulo III, como resultado se brinda una definición propia del concepto, la cual reúne los principales aspectos que los identifican (aprendizaje y diversión), igualmente, se realizó una caracterización de los juegos serios a nivel general y, en particular, se brindó una serie de características funcionales.

Las interrogantes 6, 7 y 8 relacionadas con *cloud computing*, describieron en detalle en el capítulo IV, como parte de los resultados, se identifica una serie de aspectos que caracterizan a este modelo computacional, mismas que posteriormente se tomaron como referencia para la elaboración de una definición propia de *cloud computing*. Luego, se concluye que, este modelo computacional podría potenciar el rendimiento académico, la eficacia y la eficiencia, a través de la potencialización de la colaboración y la implementación del *feedback* efectivo.

En relación con la pregunta 9 vinculada con los aspectos que se deben considerar en el desarrollo de software, se describe en detalle en el capítulo V, donde se obtuvo como resultado la identificación de las tareas que se deben realizar en el diseño de sistemas, igualmente, se destaca la programación orientada a objetos por su característica relevante en relación con la reutilización de piezas de software

En relación con las preguntas de investigación 10 y 11, las mismas se abordaron en detalle en el capítulo VI, relacionadas con metodologías utilizadas para el diseño de juegos serios, se obtuvo como resultado la elaboración de un ***checklist*** donde se resumen parte de las características distintivas o representativas identificadas que podrían ser consideradas al definir una metodología que permita guiar el diseño y desarrollo de juegos serios, mismas que

fueron tomadas en cuenta para elaborar la propuesta metodológica y arquitectónica de esta investigación.

Las interrogantes 12 y 13 vinculadas con las arquitecturas que se han utilizado para el diseño de juegos serios, se abordaron y describieron en detalle en el capítulo VII, se obtuvo como resultado la elaboración de un **checklist** donde se resumen las principales consideraciones a tomar en cuenta al diseñar una propuesta arquitectónica para el diseño y desarrollo de videojuegos, en particular, juegos serios.

Luego, en el capítulo VIII se describe en detalle la metodología INTEGRA+506 la cual es una propuesta que permite guiar la integración de aspectos metodológicos y arquitectónicos para el diseño de juegos serios orientados a potenciar en el profesorado la formación de competencias tecnológicas. En la Figura 8.1, se muestra a manera de síntesis las secuencias de dichos pasos descritos.

En el capítulo IX se presenta el diseño de un prototipo no funcional del juego serio denominado **SkillNet**, el cual se orienta a la formación de competencias tecnológicas, específicamente las relacionadas con redes de datos.

A nivel general, la metodología **INTEGRA+506** está pensada para el diseño de juegos serios orientados a la formación de competencias tecnológicas, en la cual se toma en consideración aspectos relevantes desde el *cloud computing* (principalmente, los relacionados con software como servicio), componente pedagógico y elementos propios de la gamificación. Para la validación de la metodología en sus diferentes fases, se contó con el criterio de expertos de diferentes áreas del conocimiento.

La metodología **INTEGRA+506** mostró efectividad para guiar el trabajo realizado en el diseño del prototipo no funcional de **SkillNet**. Su aplicación ayudó a ordenar todo el proceso.

Finalmente, se presenta la definición propia de varios términos, tales como competencias tecnológicas, *cloud computing* y juegos serios, mismas como resultado de la investigación teórica realizada, las cuales servirán de base para futuras investigaciones.

En la próxima sección, se citan y describen las líneas de trabajo y/o investigación a futuro, identificadas a partir de los hallazgos de la investigación realizada.

### 10.3. Líneas de trabajo futuro

Se trabajará en el diseño e implementación del prototipo funcional del juego serio **SkillNet**, se considerará la posibilidad de que el mismo sea ofrecido como un software de servicio.

Se debe trabajar en la aplicación de la propuesta metodológica **INTEGRA+506** para el diseño de prototipos de juegos serios en diferentes ámbitos educativos, así como en distintas áreas del conocimiento y competencias.

Se dará continuidad a esta temática a través de la Maestría en Redes de Datos.

### 10.4. Producción científica

La producción científica derivada de los resultados parciales de la tesis, así como la producción científica derivada durante el proceso de formación, puede ser consultada mediante el perfil del tesista creado en Google Scholar [juan.sandidelgado@ucr.c.cr](mailto:juan.sandidelgado@ucr.c.cr), mismo que se resume a continuación:

**Sandí-Delgado**, J. C., & Bazán, P. A. (03 noviembre, 2022). Arquitecturas para el Diseño de Juegos Serios. Una Revisión Sistemática de Literatura. *X Simposio de Informática Empresarial*. Turrialba, Costa Rica [Aceptado para ser publicado por la Revista InterSedes].

**Sandí-Delgado**, J. C., & Bazán, P. A. (2021). Diseño de juegos serios: Análisis de metodologías. *E-Ciencias de La Información*, 11(2), 1–24. <https://doi.org/10.15517/eci.v11i2.45505>

**Sandí-Delgado**, J. C., & Bazán, P. A. (2020). A systematic literature review of methodologies used for the design of serious games. A comparative analysis. In M. Carmo (Ed.), *International Conference on Education and New Developments* (END 2020) (pp. 358–362). <https://doi.org/10.36315/2020end076>

**Sandí-Delgado**, J. C., & Bazán, P. A. (2019). Serious Educational Games as a Service: Challenges and Solutions. *JCS&T. Journal of Computer Science & Technology*, 19(01), 66–80. <https://doi.org/10.24215/16666038.19.e07>

**Sandí-Delgado**, J. C., Sanz, C. V., & Lovos, E. N. (2022). Acceptance of Serious Games to Develop Digital Competencies in Higher Education. *The*

*Electronic Journal of E-Learning (EJEL)*, 20(3), pp351-367.  
<https://doi.org/10.34190/ejel.20.3.2181>

**Sandí-Delgado**, J. C., & Sanz, C. V. (2020). Juegos serios para potenciar la adquisición de competencias digitales en la formación del profesorado. *Revista Educación*, 44(1). <https://doi.org/10.15517/REVEDU.V44I1.37228>

**Sandí-Delgado**, J. C., & Sanz, C. V. (2018). Revisión y análisis sobre competencias tecnológicas esperadas en el profesorado en Iberoamérica. *Edutec. Revista Electrónica de Tecnología Educativa*, 66, 93–121. <https://doi.org/10.21556/edutec.2018.66.1225>

**Sandí-Delgado**, J. C., & Cruz-Alvarado, M. A. (2018). Análisis comparativo de juegos móviles educativos basados en posicionamiento. *Revista InterSedes*, 19(39), 146–170. Disponible en <https://doi.org/10.15517/isucr.v19i39.34075>

**Sandí-Delgado**, J. C., & Cruz-Alvarado, M. A. (2017a). La Simulación como Recurso Electrónico para Potenciar las Habilidades Cognitivas del Estudiantado. *Revista InterSedes*, 18(37), 1–31. Disponible en <https://doi.org/10.15517/isucr.v18i37.28646>

**Sandí-Delgado**, J. C., & Cruz-Alvarado, M. A. (2017b). Repositorios institucionales digitales: Análisis comparativo entre SEDICI (Argentina) y Kérwá (Costa Rica). *E-Ciencias de La Información*, 7(1), 1–31. Disponible en <https://doi.org/10.15517/eci.v7i1.25264>

Cruz-Alvarado, M. A., **Sandí-Delgado**, J. C., & Víquez, I. G. (2017). Diseño de situaciones educativas innovadoras como estrategia didáctica para fortalecer el proceso de enseñanza-aprendizaje. *Revista Didasc@lia: Didáctica y Educación*, 8(2), 99–116. Disponible en <http://hdl.handle.net/10915/61270>

Cruz-Alvarado, M. A., & **Sandí-Delgado**, J. C. (2017). Sistemas y Tecnologías que Facilitan el Posicionamiento Indoor. *Revista Pensamiento Actual*, 17(29), 132–144. Disponible en <https://revistas.ucr.ac.cr/index.php/pensamiento-actual/article/view/31585>

**Sandí-Delgado**, J. C., & Cruz-Alvarado, M. A. (2016). Propuesta metodológica de enseñanza y aprendizaje para innovar la educación superior. *Revista InterSedes*, 17(36), 1–38. Disponible en <https://doi.org/10.15517/isucr.v17i36.27100>

# ANEXOS

# REFERENCIAS

- Abt, C. C. (1970). *Serious Games*. Viking Press.
- Aguaded, I., & Cabero, J. (2014). Avances y retos en la promoción de la innovación didáctica con las tecnologías emergentes e interactivas. *Educar, Especial 30 Aniversario*, 67–83. <https://doi.org/10.5565/rev/educar.691>
- Albert, M., & Torres, M. V. (2022a). *Desarrollo de una aplicación web como catálogo de buenas prácticas de accesibilidad*. Universitat Politècnica de València.
- Albert, M., & Torres, M. V. (2022b). *Desarrollo de una aplicación web como catálogo de buenas prácticas de accesibilidad* [Universitat Politècnica de València]. <http://hdl.handle.net/10251/188224>
- All, A., Nuñez, E. P., & Van Looy, J. (2016). Assessing the effectiveness of digital game-based learning: Best practices. *Computers & Education*, 92–93(January–February), 90–103. <https://doi.org/10.1016/j.compedu.2015.10.007>
- Almerich, G., Orellana, N., Suárez-Rodríguez, J., & Díaz-García, I. (2016). Teachers' information and communication technology competences: A structural approach. *Computers & Education*, 100(September), 110–125. <https://doi.org/10.1016/j.compedu.2016.05.002>
- Alshamrani, A., & Bahattab, A. (2010). Transformation from Data Flow Diagram to UML2.0 activity diagram. *2010 IEEE International Conference on Progress in Informatics and Computing*, 2, 1010–1014. <https://doi.org/10.1109/PIC.2010.5687483>
- Alshamrani, A., & Bahattab, A. (2015). A Comparison Between Three SDLC Models Waterfall Model , Spiral Model , and Incremental / Iterative Model. *International Journal of Computer Science*, 12(1), 106–111. <https://ijcsi.org/papers/IJCSI-12-1-1-106-111.pdf>
- Alshazly, A. A., Elfatatty, A. M., & Abougabal, M. S. (2014). Detecting defects in software requirements specification. *Alexandria Engineering Journal*, 53(3), 513–527. <https://doi.org/10.1016/j.aej.2014.06.001>
- Ammann, P., & Offutt, J. (2008). *Introducción to Software Testing* (1st ed.). Cambridge university press. <https://epdf.pub/introduction-to-software-testing.html>
- Angeli, C., & Valanides, N. (2005). Preservice elementary teachers as information and communication technology designers: an instructional systems design model based on an expanded view of pedagogical content knowledge. *Journal of Computer Assisted Learning*, 21(4), 292–302. <https://doi.org/10.1111/j.1365-2729.2005.00135.x>
- Angeli, C., & Valanides, N. (2009). Epistemological and methodological issues for the conceptualization, development, and assessment of ICT-TPCK: Advances in technological pedagogical content knowledge (TPCK). *Computers & Education*, 52(1), 154–168. <https://doi.org/10.1016/j.compedu.2008.07.006>
- Angelini, M. L., García-Carbonell, A., & Martínez-Alzamora, N. (2017). Estudio de correlación entre la simulación telemática y las destrezas lingüísticas en inglés. *REDIE. Revista Electrónica de Investigación Educativa*, 19(1), 141–156.

- <https://doi.org/10.24320/redie.2017.19.1.1100>
- Anwer, F., Aftab, S., Waheed, U., & Muhammad, S. (2017). Agile Software Development Models TDD , FDD , DSDM , and Crystal Methods: A Survey. *INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING*, 8(2), 1–10.  
[https://www.researchgate.net/publication/316273992\\_Agile\\_Software\\_Development\\_Models\\_TDD\\_FDD\\_DSDM\\_and\\_Crystal\\_Methods\\_A\\_Survey](https://www.researchgate.net/publication/316273992_Agile_Software_Development_Models_TDD_FDD_DSDM_and_Crystal_Methods_A_Survey)
- Apezteguía, M., Rapetti, D. E., Gordillo, S. E., & Challiol, C. (2014). *Juego Educativo Móvil Colaborativo* [Universidad Nacional de La Plata (UNLP)].  
<http://hdl.handle.net/10915/47078>
- Archambault, L. M., & Barnett, J. H. (2010). Revisiting technological pedagogical content knowledge: Exploring the TPACK framework. *Computers & Education*, 55(4), 1656–1662. <https://doi.org/10.1016/j.compedu.2010.07.009>
- Archuby, F. H., Sanz, C. V., & Pesado, P. M. (2017a). Desafiate: Juego Serio para la Autoevaluación de los Alumnos y su Integración con un Entorno Virtual de Enseñanza y Aprendizaje. *XXIII Congreso Argentino de Ciencias de La Computación*, 284–294. <http://hdl.handle.net/10915/63543>
- Archuby, F. H., Sanz, C. V., & Pesado, P. M. (2017b). Juego serio como actividad de autoevaluación de los alumnos y su integración con un entorno virtual de enseñanza y aprendizaje [Universidad Nacional de La Plata (UNLP)]. In *XXIII Congreso Argentino de Ciencias de la Computación*.  
<http://hdl.handle.net/10915/59652>
- Archuby, F. H., Sanz, C. V., & Pesado, P. M. (2017c). *Juego serio como actividad de autoevaluación de los alumnos y su integración con un entorno virtual de enseñanza y aprendizaje* [Tesis presentada para obtener el grado de Licenciatura en Sistemas. Universidad Nacional de La Plata (UNLP)].  
<http://hdl.handle.net/10915/59652>
- Archuby, F. H., Sanz, C. V., & Pesado, P. M. (2018). Desafiate: juego serio para la autoevaluación. *XIII Congreso de Tecnología En Educación y Educación En Tecnología*, 200, 209–212. <http://sedici.unlp.edu.ar/handle/10915/69074>
- Arevalos, V. (2014). La formación del profesorado universitario en el uso de las tecnologías de la información y comunicación (TIC) en la Universidad Nacional de Itapúa – Facultad de Humanidades y la Universidad Autónoma de Encarnación (UNAE). In J. Asenjo, Ó. Macías, & J. C. Toscano (Eds.), *Actas del Congreso Iberoamericano de Ciencia, Tecnología, Innovación y Educación* (pp. 1–10). Organización de Estados Iberoamericanos (OEI).  
<http://www.oei.es/historico/congreso2014/21memorias2014.php>
- Armendariz, I. E., Ecclesia, S. G., Gordillo, S. E., & Challiol, C. (2015). *Implementación de un Prototipo de Aplicación Web Móvil Sensible al Contexto* [Universidad Nacional de La Plata (UNLP)]. <http://hdl.handle.net/10915/48097>
- Aurum, A., & Wohlin, C. (2005). Engineering and managing software requirements. In A. Aurum & C. Wohlin (Eds.), *Engineering and Managing Software Requirements* (1st ed.). Springer. <https://doi.org/10.1007/3-540-28244-0>

- Aybar, V. del C., Queiruga, C., & Banchoff, C. (2012). *Aplicaciones complementarias a ROBOCODE que faciliten el aprendizaje de programación en escuelas secundarias* (Issue Plan 90). <http://hdl.handle.net/10915/47050>
- Baca, A. R. (2015). Competencias docentes digitales: propuesta de un perfil. *Píxel-Bit. Revista de Medios y Educación*, 46(Enero), 235–248. <http://www.redalyc.org/articulo.oa?id=36832959015>
- Bazán, P. A. (2015). *Implementación de procesos de negocio a través de servicios aplicando metamodelos, software distribuido y aspectos sociales* [Tesis presentada para obtener el grado de Doctor en Ciencias Informáticas. Universidad Nacional de La Plata (UNLP)]. <http://hdl.handle.net/10915/45100>
- Bazán, P. A., Fernández, A., Del Río, N., Molinari, L., Pérez, J. P., & Banchoff, M. (2017). *Aplicaciones, servicios y procesos distribuidos: Una visión para la construcción de software* (P. A. Bazán (ed.); 1st ed.). Editorial de la Universidad Nacional de La Plata (EDULP). <http://hdl.handle.net/10915/62354>
- Bezanilla, M. J., Arranz, S., Rayón, A., Rubio, I., Menchaca, I., Guenaga, M., & Aguilar, E. (2014). A proposal for generic competence assessment in a serious game. *NAER. Journal of New Approaches in Educational Research*, 3(1), 42–51. <https://doi.org/10.7821/naer.3.1.42-51>
- Bhatia, L., & Prasad, S. S. (2015). COPAL - Cognitive Personalized Aid for Learning. *International Conference on Cognitive Computing and Information Processing(CCIP)*, 1–6. <https://doi.org/10.1109/CCIP.2015.7100698>
- Biletskiy, Y., Baghi, H., Keleberda, I., & Fleming, M. (2009). An adjustable personalization of search and delivery of learning objects to learners. *Expert Systems with Applications*, 36(5), 9113–9120. <https://doi.org/10.1016/j.eswa.2008.12.038>
- Bione, J., Miceli, P., Sanz, C. V., & Artola, V. (2017a). *AstroCódigo. Un juego serio para la introducción de jóvenes en los conceptos básicos de la programación*. <http://hdl.handle.net/10915/61204>
- Bione, J., Miceli, P., Sanz, C. V., & Artola, V. (2017b). Enseñanza de la programación con astrocódigo. *XII Congreso de Tecnología En Educación y Educación En Tecnología (TE&ET)*, 454–455. <http://hdl.handle.net/10915/63453>
- Blackwell, C. K., Lauricella, A. R., & Wartella, E. (2016). The influence of TPACK contextual factors on early childhood educators' tablet computer use. *Computers & Education*, 98(July), 57–69. <https://doi.org/10.1016/j.compedu.2016.02.010>
- Boulahrouz, M., Medir, R. M., & Calabuig, S. (2019). Tecnologías digitales y educación para el desarrollo sostenible. Un análisis de la producción científica. *Píxel-Bit. Revista de Medios y Educación*, 54, 83–106. <https://doi.org/10.12795/pixelbit.2019.i54.05>
- Boyle, E. A., Connolly, T. M., & Hainey, T. (2011). The role of psychology in understanding the impact of computer games. *Entertainment Computing*, 2(2), 69–74. <https://doi.org/10.1016/j.entcom.2010.12.002>

## REFERENCIAS

- Boyle, E. A., Connolly, T. M., Hainey, T., & Boyle, J. M. (2012). Engagement in digital entertainment games: A systematic review. *Computers in Human Behavior*, 28(3), 771–780. <https://doi.org/10.1016/j.chb.2011.11.020>
- Boyle, E. A., Hainey, T., Connolly, T. M., Gray, G., Earp, J., Ott, M., Lim, T., Ninaus, M., Ribeiro, C., & Pereira, J. (2016). An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games. *Computers & Education*, 94, 178–192. <https://doi.org/10.1016/j.compedu.2015.11.003>
- Bozu, Z., & Canto, P. J. (2009). El profesorado universitario en la sociedad del conocimiento: competencias profesionales docentes. *Revista de Formación e Innovación Educativa Universitaria*, 2(2), 87–97. <http://tecnologiaedu.us.es/mec2011/htm/mas/3/31/26.pdf>
- Brown, J. (2005). Incorporación de las tecnologías de información y comunicación en la docencia universitaria estatal costarricense: problemas y soluciones. *Revista Electrónica Actualidades Investigativas En Educación*, 5(1), 1–21. <https://doi.org/10.15517/aie.v5i1.9118>
- Buendía-García, F., García-Martínez, S., Navarrete-Ibañez, E. M., & Cervelló-Donderis, M. J. (2013). Designing Serious Games for getting transferable skills in training settings. *Interaction Design and Architecture(s) Journal - IxD&A*, 19, 47–62. <http://hdl.handle.net/10251/47743>
- Butler, G., Grogono, P., Shinghal, R., & Tjandra, I. (1995). Analyzing the Logical Structure of Data Flow Diagrams in Software Documents. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 575–578.
- Cabarkapa, D. (2015). *Application of Cisco Packet Tracer 6.2 in teaching of advanced computer networks*. DataCite. <https://doi.org/10.13140/RG.2.1.4881.6802>
- Cabero, J. (2010). Los retos de la integración de las TICs en los procesos educativos. Límites y posibilidades. *Perspectiva Educacional*, 49(1), 32–61. <https://doi.org/10.4151/07189729-Vol.49-Iss.1-Art.3>
- Cabero, J., Marín, V., & Castaño, C. (2015). Validación de la aplicación del modelo TPACK para la formación del profesorado en TIC. *@Tic. Revista D'Innovació Educativa*, 14, 13–22. <https://doi.org/10.7203/attic.14.4001>
- Cabrera, J. (2017). *Modular Design Frameworks: A Projects-based Guide for UI/UX Designers* (K. Jameson (ed.); 1st ed.). Apress. <https://doi.org/10.1007/978-1-4842-1688-0>
- Calabor, M. S., Mora, A., & Moya, S. (2017). Adquisición de competencias a través de juegos serios en el área contable: un análisis empírico. *Revista de Contabilidad*, 7, 1–10. <https://doi.org/10.1016/j.rctsar.2016.11.001>
- Calderón, A., & Ruiz, M. (2015). A systematic literature review on serious games evaluation: An application to software project management. *Computers & Education*, 87(September), 396–422. <https://doi.org/10.1016/j.compedu.2015.07.011>

- Campo, M. F., De Cabrales, R. S., Martínez, P. del P., Rendón, H. J., & Calderón, G. G. (2013). Competencias TIC Para el Desarrollo Profesional Docente. In *Ministerio de Educación Nacional*. Ministerio de Educación Nacional de Colombia. <https://doi.org/10.1017/CBO9781107415324.004>
- Campos, J., Brenes, O. L., & Solano, A. (2010). Competencias del Docente de Educación Superior en Línea. *Revista Electrónica Actualidades Investigativas En Educación*, 10(3), 1–19. <https://doi.org/10.15517/aie.v10i3.10141>
- Cano, S. P. (2016). *Propuesta Metodológica para el Diseño de Juegos Serios para Niños con Implante Coclear* [(Tesis doctoral)]. Universidad del Cauca]. [http://www.unicauca.edu.co/doctoradoe/publicaciones/Monografia\\_Cano.pdf](http://www.unicauca.edu.co/doctoradoe/publicaciones/Monografia_Cano.pdf)
- Cano, S. P., Muñoz, J., Collazos, C. A., González, C. S., & Zapata, S. (2016). Toward a methodology for serious games design for children with auditory impairments. *IEEE Latin America Transactions*, 14(5), 2511–2521. <https://doi.org/10.1109/TLA.2016.7530453>
- Canós, J. H., Letelier, P., & Panadés, M. C. (2003). Métodologías Ágiles en el Desarrollo de Software. In P. L. Torres & E. A. S. López (Eds.), *VIII Jornadas de Ingeniería del Software y Bases de Datos* (pp. 1–8). Ingeniería del Software y Sistemas de Información. <https://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>
- Cárdenas-García, P. J., Pulido-Fernández, J. I., & Carrillo-Hidalgo, I. (2016). Adquisición de competencias en el Grado de Turismo mediante el aprendizaje basado en estudios de caso. *Aula Abierta*, 44(1), 15–22. <https://doi.org/10.1016/j.aula.2014.12.001>
- Carrión, M., Quispi, B., Lema, S., Santórum, M., & Aguilar, J. (2019). Creando un juego serio educativo mediante un enfoque de diseño centrado en el usuario. *Revista Iberica de Sistemas e Tecnologias de Informação*, E23, 158–170. <https://doi.org/10.17013/risti.n.pi-pf>
- Carvalho, M. B., Bellotti, F., Berta, R., De Gloria, A., Gazzarata, G., Hu, J., & Kickmeier-Rust, M. (2015). A case study on Service-Oriented Architecture for Serious Games. *Entertainment Computing*, 6, 1–10. <https://doi.org/10.1016/j.entcom.2014.11.001>
- Carvalho, M. B., Bellotti, F., Berta, R., De Gloria, A., Islas, C., Baalsrud, J., Hu, J., & Rauterberg, M. (2015). An activity theory-based model for serious games analysis and conceptual design. *Computers & Education*, 87, 166–181. <https://doi.org/10.1016/j.compedu.2015.03.023>
- Carvalho, M. B., Bellotti, F., Hu, J., Hauge, J. B., Berta, R., De Gloria, A., & Rauterberg, M. (2015). Towards a service-oriented architecture framework for educational serious games. *IEEE 15th International Conference on Advanced Learning Technologies*, 147–151. <https://doi.org/10.1109/ICALT.2015.145>
- Castells, M. (2009). La apropiación de las tecnologías. La cultura juvenil en la era digital. *Telos: Cuadernos de Comunicación e Innovación*, 81, 111–113. <https://dialnet.unirioja.es/servlet/articulo?codigo=3113585&info=resumen&idioma=SPA>

- Cataldi, Z., Lage, F., & Cabero, J. (2010). La promoción de competencias en el trabajo grupal con base en tecnologías informáticas y sus implicancias didácticas. *Píxel-Bit. Revista de Medios y Educación, julio-diciembre(37)*, 209–224. [https://www.academia.edu/30066593/La\\_Promoción\\_De\\_Competencias\\_en\\_El\\_Trabajo\\_Grupal\\_Con\\_Base\\_en\\_Tecnologías\\_Informáticas\\_y\\_Sus\\_Implicancias\\_Didácticas\\_Promotion\\_of\\_Competences\\_in\\_Groupal\\_Work\\_Based\\_on\\_Informatic\\_Technologies\\_and\\_This\\_Didactic\\_Consideratio](https://www.academia.edu/30066593/La_Promoción_De_Competencias_en_El_Trabajo_Grupal_Con_Base_en_Tecnologías_Informáticas_y_Sus_Implicancias_Didácticas_Promotion_of_Competences_in_Groupal_Work_Based_on_Informatic_Technologies_and_This_Didactic_Consideratio)
- Chang, K.-E., Wu, L.-J., Weng, S.-E., & Sung, Y.-T. (2012). Embedding game-based problem-solving phase into problem-posing system for mathematics learning. *Computers & Education, 58*(2), 775–786. <https://doi.org/10.1016/j.compedu.2011.10.002>
- Chen, R.-J. (2010). Investigating models for preservice teachers' use of technology to support student-centered learning. *Computers & Education, 55*(1), 32–42. <https://doi.org/10.1016/j.compedu.2009.11.015>
- Chipia, J. F. (2011). Juegos Serios: Alternativa Innovadora. *Conocimiento Libre y Educación (CLED), 2*(2), 1–18. <http://erevistas.saber.ula.ve/index.php/cled/article/view/4862>
- Choi, M., Cristol, D., & Gimbert, B. (2018). Teachers as digital citizens: The influence of individual backgrounds, internet use and psychological characteristics on teachers' levels of digital citizenship. *Computers & Education, 121*, 143–161. <https://doi.org/10.1016/j.compedu.2018.03.005>
- Clear, T., Beecham, S., Daniels, M., Oudshoorn, M., Barr, J., & Noll, J. (2016). Developments in Global Software Engineering Education. *IEEE Frontiers in Education Conference (FIE)*, 1–4. <https://doi.org/10.1109/FIE.2016.7757471>
- Clements, P., & Northrop, L. (2001). *Software Product Lines: Practices and Patterns* (3rd ed.). Addison-Wesley Professional. <https://doi.org/10.1109/ms.2002.1020300>
- Cobeña, J. A. (2021). *Plataformas de computación en la nube para el desarrollo de aplicaciones web* [Pontificia Universidad Católica del Ecuador Sede Esmeraldas]. <https://repositorio.pucese.edu.ec/handle/123456789/2595>
- Cobo, J. C. (2009). El concepto de tecnologías de la información. Benchmarking sobre las definiciones de las TIC en la sociedad del conocimiento. *Zer - Revista de Estudios de Comunicación, 14*(27), 295–318. <https://doi.org/10.4067/S0718-13372003000200001>
- Connolly, T. M., Boyle, E. A., MacArthur, E., Hainey, T., & Boyle, J. M. (2012). A systematic literature review of empirical evidence on computer games and serious games. *Computers & Education, 59*(2), 661–686. <https://doi.org/10.1016/j.compedu.2012.03.004>
- Costa, T. K. L., Machado, L. S., Valenca, A. M. G., & Moraes, R. M. (2016). Architecture to portals of serious games and virtual environments with performance evaluation during sequences of activities. *IEEE International Conference on Serious Games and Applications for Health (SeGAH)*, 1–8. <https://doi.org/10.1109/SeGAH.2016.7586225>

- Cruz-Alvarado, M. A., & Bazán, P. A. (2018). Understanding the Internet of Nano Things: overview, trends, and challenges. *E-Ciencias de La Información*, 9(1), 1–30. <https://doi.org/10.15517/eci.v1i1.33807>
- Cruz-Alvarado, M. A., Sandí-Delgado, J. C., & Víquez-Barrantes, I. G. (2017). Diseño de situaciones educativas innovadoras como estrategia didáctica para fortalecer el proceso de enseñanza-aprendizaje. *Didasc@lia: Didáctica y Educación*, 8(2), 99–116. <https://200.14.53.93/index.php/didascalia/article/view/611>
- Cruz-Pérez, M. A., Pozo-Vinueza, M. A., Andino-Jaramillo, A. F., & Arias-Parra, A. D. (2019). Las Tecnologías de la Información y la Comunicación (TIC) como forma investigativa interdisciplinaria con un enfoque intercultural para el proceso de formación de los estudiantes. *E-Ciencias de La Información*, 9(1), 1–15. <https://doi.org/10.15517/eci.v1i1.33052>
- Cruz-Rodríguez, E. D. carmen. (2019). Importancia del manejo de competencias tecnológicas en las prácticas docentes de la Universidad Nacional Experimental de la Seguridad (UNES). *Revista Educación*, 43(1), 196–218. <https://doi.org/10.15517/revedu.v43i1.27120>
- Cuberos de Quintero, M. A., & Vivas, M. (2017). Relación entre didáctica, gerencia y el uso educativo de las TIC. *Revista Electrónica Actualidades Investigativas En Educación*, 17(1), 1–31. <https://doi.org/10.15517/aie.v17i1.27198>
- Cuevas-Cordero, F., & García-Fallas, J. (2014). Las TIC en la formación docente. In J. Asenjo, Ó. Macías, & J. C. Toscano (Eds.), *Actas del Congreso Iberoamericano de Ciencia, Tecnología, Innovación y Educación* (pp. 1–29). Organización de Estados Iberoamericanos (OEI). <http://www.oei.es/historico/congreso2014/21memorias2014.php>
- Dalal, M., Archambault, L. M., & Shelton, C. (2017). Professional Development for International Teachers: Examining TPACK and Technology Integration Decision Making. *JRTE. Journal of Research on Technology in Education*, 49(3–4), 117–133. <https://doi.org/10.1080/15391523.2017.1314780>
- de Oliveira, I., & Senger de Souza, S. do R. (2019). Study and definition of project attributes for selection of testing techniques for concurrent software. *Estendidos Da Conferência Brasileira de Software: Teoria e Prática*, 24–30. [https://doi.org/10.5753/cbsoft\\_estendido.2019.7652](https://doi.org/10.5753/cbsoft_estendido.2019.7652)
- DEBORAH Game. (2017). *DEBORAH - Double Entry bookkeeping or accounting history*. DEBORAH Game. <http://deborahahg.wixsite.com/deborah/about>
- Del-Moral, M.-E., & Guzmán-Duque, A.-P. (2014). CityVille: collaborative game play, communication and skill development in social networks. *NAER. Journal of New Approaches in Educational Research*, 3(1), 11–19. <https://doi.org/10.7821/naer.3.1.11-19>
- Díaz-Erazo, A. D., Morales-Morales, R. M., Pineda-Chávez, V. K., & Morales-Cardoso, L. S. (2022). Comparative Analysis of performance for SQL and NoSQL Databases. *17th Iberian Conference on Information Systems and Technologies*, 1–14. <https://doi.org/10.23919/CISTI54924.2022.9820292>

- Díaz, F. J., Queiruga, C., & Fava, L. (2015). Juegos Serios y Educación. *XVII Workshop de Investigadores En Ciencias de La Computación*, 1–5. <http://hdl.handle.net/10915/46458>
- Ducasse, S., & Pollet, D. (2017). *Learning Object-Oriented Programming, Design and TDD with Pharo*. <https://ci.inria.fr/pharo-contribution/view/Books/job/LearningObjectOrientedProgramming/98/artifact/book.pdf>
- El Mawas, N. (2014). An architecture for co-designing participatory and knowledge-intensive serious games: ARGILE. *International Conference on Collaboration Technologies and Systems (CTS)*, 387–394. <https://doi.org/10.1109/CTS.2014.6867593>
- Elnaggar, A., & Reichardt, D. (2016). Digitizing the hand rehabilitation using serious games methodology with user-centered design approach. In H. R. Arabnia, L. Deligiannidis, & M. Yang (Eds.), *International Conference on Computational Science and Computational Intelligence, CSCI* (pp. 13–22). IEEE. <https://doi.org/10.1109/CSCI.2016.0011>
- Evans, F., & Massa, S. M. (2017). Elicitación y Especificación de Requerimientos en Pervasive Serious Games. *XIX Workshop de Investigadores En Ciencias de La Computación*, 1199–1203. <http://hdl.handle.net/10915/62883>
- Evans, F., Spinelli, A. T., Zapirain, E. A., Massa, S. M., & Soriano, F. (2016). Proceso de desarrollo de Serious Games. Diseño centrado en el usuario, jugabilidad e inmersión. *III Congreso Argentino de Ingeniería - IX Congreso de Enseñanza de La Ingeniería*, 1–13. [https://www.researchgate.net/publication/326156974\\_proceso\\_de\\_desarrollo\\_de\\_serious\\_games\\_diseño\\_centrado\\_en\\_el\\_usuario\\_jugabilidad\\_e\\_inmersion](https://www.researchgate.net/publication/326156974_proceso_de_desarrollo_de_serious_games_diseño_centrado_en_el_usuario_jugabilidad_e_inmersion)
- Farjon, D., Smits, A., & Voogt, J. M. (2019). Technology integration of pre-service teachers explained by attitudes and beliefs, competency, access, and experience. *Computers & Education*, 130, 81–93. <https://doi.org/10.1016/j.compedu.2018.11.010>
- Ferrari, A. (2013). DIGCOMP: A Framework for Developing and Understanding Digital Competence in Europe. In Y. Punie & B. N. Brečko (Eds.), *JRC scientific and policy reports*. Publications Office of the European Union. <https://doi.org/10.2788/52966>
- Flores, C. E., López, M. Q., Orozco, H. R., & Pérez, I. R. (2019). Arquitectura de un juego serio inteligente basado en retos de matemáticas básicas. *ReCIBE. Revista Electrónica de Computación, Informática, Biomédica y Electrónica*, 8(2), 1–14. <https://doi.org/10.32870/recibe.v8i2.135>
- Fowler, M., & Highsmith, J. (2001). The Agile Manifesto. *Software Development*, 8(9), 28–35. <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>
- Fragoso-Díaz, O. G., Santaolaya-Salgado, R., & De Gyves-Avila, S. (2008). Web Services for Software Development: The Case of a Web Service That Composes Web Services. *The Third International Conference on Software Engineering Advances*, 31–36. <https://doi.org/10.1109/ICSEA.2008.65>

- Fraillon, J., Ainley, J., Schulz, W., Friedman, T., & Gebhardt, E. (2014). *Preparing for life in a digital age. The IEA international computer and information literacy study international report*. Springer.
- Frécon, E., & Stenius, M. (1998). DIVE: A scaleable network architecture for distributed virtual environments. *Distributed Systems Engineering*, 5(3), 91–100. <https://doi.org/10.1088/0967-1846/5/3/002>
- Freeman, P., & Hart, D. (2004). A science of design for software-intensive systems. *Communications of the ACM - Interactive Immersion in 3D Graphic*, 47(8), 19–21. <https://doi.org/10.1145/1012037.1012054>
- Frezzo, D. C., Behrens, J. T., Mislevy, R. J., West, P., & DiCerbo, K. E. (2009). Psychometric and Evidentiary Approaches to Simulation Assessment in Packet Tracer Software. *Fifth International Conference on Networking and Services (ICNS)*, 555–560. <https://doi.org/10.1109/ICNS.2009.89>
- Gagliardi, V. (2021). Decoupled Django: Understand and Build Decoupled Django Architectures for JavaScript Front-ends. In J. Markham (Ed.), *Decoupled Django*. Apress. <https://doi.org/10.1007/978-1-4842-7144-5>
- García-Fallas, J. (2004). *Ambientes con recursos tecnológicos: Escenarios para la construcción de procesos pedagógicos* (1st ed.). Editorial Universidad Estatal a Distancia (UNED).
- Garita, F. A., Lizano, F., & Cordero, C. M. (2019). Metodologías para el Desarrollo de Videojuegos Serios: Una Revisión de Literatura. *Tecnología Educativa Revista CONAIC*, 6(1), 103–114. <https://doi.org/10.32671/terc.v6i1.85>
- Garlan, D., & Shaw, M. (1994). *An Introduction to Software Architecture* (Carnegie M). DEFENSE TECHNICAL INFORMATION CENTER. [https://doi.org/10.1142/9789812798039\\_0001](https://doi.org/10.1142/9789812798039_0001)
- Giannakos, M. N. (2013). Enjoy and learn with educational games: Examining factors affecting learning performance. *Computers & Education*, 68(October), 429–439. <https://doi.org/10.1016/j.compedu.2013.06.005>
- Girard, C., Ecale, J., & Magnan, A. (2013). Serious games as new educational tools: How effective are they? A meta-analysis of recent studies. *Journal of Computer Assisted Learning*, 29(3), 207–219. <https://doi.org/10.1111/j.1365-2729.2012.00489.x>
- Gómez-Fuentes, M. del C. (2011). *Notas Del Curso Análisis De Requerimientos* (M. del C. Gómez (ed.); 1st ed.). Universidad Autónoma Metropolitana, Unidad Cuajimalpa. [http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas\\_Analisis\\_Requerimiento.pdf](http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas_Analisis_Requerimiento.pdf)
- González, D. (2017). Ambientes colaborativos virtuales para el aprendizaje individual. *Revista Electrónica Actualidades Investigativas En Educación*, 17(2), 1–29. <https://doi.org/10.15517/aie.v17i1.28092>
- Graham, C. R. (2011). Theoretical considerations for understanding technological pedagogical content knowledge (TPACK). *Computers & Education*, 57(3),

- 1953–1960. <https://doi.org/10.1016/j.compedu.2011.04.010>
- Granollers, T. (2004). *MPiu+a. Una Metodología que integra la ingeniería del software, la interacción persona-ordenador y la accesibilidad en el contexto de equipos de desarrollo multidisciplinares* [(Tesis doctoral)]. Universitat de Lleida]. <https://www.tdx.cat/handle/10803/8120#page=1>
- Greenhalgh, C., & Benford, S. (1995). MASSIVE: A Collaborative Virtual Environment for Teleconferencing. *ACM Transactions on Computer-Human Interaction*, 2(3), 239–261. <https://doi.org/10.1145/210079.210088>
- Guenaga, M., Eguíluz, A., Rayón, A., & Quevedo, E. (2015). Un juego Serio para Desarrollar y Evaluar la Competencia de Trabajo en Equipo. *Revista Iberoamericana de Informática Educativa*, 21(Enero-Junio), 3–11. <http://dialnet.unirioja.es/servlet/articulo?codigo=5113264>
- Guillén-Nieto, V., & Aleson-Carbonell, M. (2012). Serious games and learning effectiveness: The case of It's a Deal! *Computers & Education*, 58(1), 435–448. <https://doi.org/10.1016/j.compedu.2011.07.015>
- Gutiérrez-Hernández, R. E., Álvarez, F. J., & Muñoz-Arteaga, J. (2013). *Arquitectura de Software para Juegos Serios con Aspectos Culturales: Caso de Estudio en un Videojuego para Formulas Temperatura*. [https://www.researchgate.net/publication/236162869\\_Arquitectura\\_de\\_Software\\_para\\_Juegos\\_Serios\\_con\\_Aspectos\\_Culturales\\_Caso\\_de\\_Estudio\\_en\\_un\\_Videojuego\\_para\\_Formulas\\_Temperatura](https://www.researchgate.net/publication/236162869_Arquitectura_de_Software_para_Juegos_Serios_con_Aspectos_Culturales_Caso_de_Estudio_en_un_Videojuego_para_Formulas_Temperatura)
- Hainey, T., Connolly, T. M., Boyle, E. A., Wilson, A., & Razak, A. (2016). A systematic literature review of games-based learning empirical evidence in primary education. *Computers & Education*, 102(November), 202–223. <https://doi.org/10.1016/j.compedu.2016.09.001>
- Häkkinen, P., Bluemink, J., Juntunen, M., & Laakkonen, I. (2012). Multiplayer 3D game in supporting team-building activities in a work organization. *12th IEEE International Conference on Advanced Learning Technologies (ICALT)*, 430–432. <https://doi.org/10.1109/ICALT.2012.242>
- Hardy, S., Dutz, T., Wiemeyer, J., Göbel, S., & Steinmetz, R. (2015). Framework for personalized and adaptive game-based training programs in health sport. *Multimedia Tools and Applications*, 74(14), 5289–5311. <https://doi.org/10.1007/s11042-014-2009-z>
- Hassan-Montero, Y., & Ortega-Santamaría, S. (2009). *Informe APEI sobre Usabilidad* (R. Lavandera-Fernández (ed.); 1st ed.). Gijón: Asociación Profesional de Especialistas en Información. <https://www.apei.es/wp-content/uploads/2013/11/InformeAPEI-Usabilidad.pdf>
- Hernández-Doria, C. A., Gómez, M. G., & Balderas, M. (2014). Inclusión de las tecnologías para facilitar los procesos de enseñanza-aprendizaje en ciencias naturales. *Revista Electrónica Actualidades Investigativas En Educación*, 14(3), 1–19. <https://doi.org/10.15517/aie.v14i3.16097>
- Hernández, C. A., Arévalo, M. A., & Gamboa, A. A. (2016). Competencias TIC para el desarrollo profesional docente en educación básica. *Praxis & Saber. Revista*

- de *Investigación y Pedagogía*, 7(14), 41–69.  
<https://doi.org/10.19053/22160159.5217>
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. del P. (2014). Análisis de datos cuantitativos. In *Metodología de la investigación* (6th ed.). Mc Graw Hill. <https://www.uca.ac.cr/wp-content/uploads/2017/10/Investigacion.pdf>
- Holland, D. D., & Piper, R. T. (2016). A technology integration education (TIE) model for millennial preservice teachers: Exploring the canonical correlation relationships among attitudes, subjective norms, perceived behavioral controls, motivation, and technological, pedagogical, and content. *JRTE. Journal of Research on Technology in Education*, 48(3), 212–226.  
<https://doi.org/10.1080/15391523.2016.1172448>
- Holloway, D. (2009). *NoviCraft*. NoviCraft: Virtual World Team Building.  
<http://www.creativeshed.com/2009/02/novicraft-virtual-world-team-building/>
- Hueso-Ibáñez, L. (2015). *Administración de Sistemas Gestores de Bases de Datos* (2nd ed.). RA-Ma.
- Hunt, K. P. (1979). An introduction to structured programming. *Behavior Research Methods & Instrumentation*, 11(2), 229–233.  
<https://doi.org/10.3758/BF03205654>
- Hussein, B. A., & Ravna, R. (2015). A template for building adaptable project risk management games. *8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, 2(September), 535–542. <https://doi.org/10.1109/IDAAKS.2015.7341363>
- IEEE Computer Society. (1990). IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology. *ieee*, 121990(1), 1.  
<https://doi.org/10.1109/IEEESTD.1990.101064>
- IMS, G. L. C. (2003). *IMS Learning Design Information Model - Version 1.0 Final Specification*. IMS Global Learning Consortium.  
[http://www.imsglobal.org/learningdesign/ldv1p0/imslid\\_infov1p0.html](http://www.imsglobal.org/learningdesign/ldv1p0/imslid_infov1p0.html)
- INET. (2016). *Programación Orientada a Objetos*.  
<http://ceaer.edu.ar/wp-content/uploads/2018/04/Apunte-Teorico-de-Programacion-OO.pdf>
- INTEF. (2017). *Marco Común de Competencia Digital Docente*. Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF).  
<http://educalab.es/documents/10180/12809/Marco+competencia+digital+docente+2017/afb07987-1ad6-4b2d-bdc8-58e9faeccea>
- Isaias, P., & Issa, T. (2015). *High Level Models and Methodologies for Information Systems*. Springer. <https://doi.org/10.1007/978-1-4614-9254-2>
- Ismail, H. (2018). WikiRec: A personalized content recommendation framework to support informal learning in wikis. *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, 273–276.  
<https://doi.org/10.1145/3209219.3213594>
- Ismail, H., & Belkhouche, B. (2019). A Reusable Software Architecture for

- Personalized Learning Systems. *International Conference on Innovations in Information Technology (IIT)*, 105–110.  
<https://doi.org/10.1109/INNOVATIONS.2018.8605997>
- ISO/IEC. (1999). *ISO 13407: Human Centred Design Process for Interactive Systems*. <http://www.ash-consulting.com/ISO13407.pdf>
- Janitor, J., Jakab, F., & Kniewald, K. (2010). Visual Learning Tools for Teaching/Learning Computer Networks: Cisco Networking Academy and Packet Tracer. *Sixth International Conference on Networking and Services*, 351–355.  
<https://doi.org/10.1109/ICNS.2010.55>
- Jansen, A., & Bosch, J. (2005). Software Architecture as a Set of Architectural Design Decisions. *5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05)*. <https://doi.org/10.1109/WICSA.2005.61>
- Janssen, J., Surendralal, S., Lysogorskiy, Y., Todorova, M., Hickel, T., Drautz, R., & Neugebauer, J. (2019). pyiron: An integrated development environment for computational materials science. *Computational Materials Science*, 163, 24–36.  
<https://doi.org/10.1016/j.commatsci.2018.07.043>
- Jepp, P., Fradinho, M., & Pereira, J. M. (2010). An agent framework for a modular serious game. In K. Debattista, M. Dickey, A. Proen  a, & L. P. Santos (Eds.), *Second International Conference on Games and Virtual Worlds for Serious Applications* (pp. 19–26). <https://doi.org/10.1109/VS-GAMES.2010.25>
- Jones, C. B. (2020). *Understanding Programming Languages*. Springer Nature Switzerland. <https://doi.org/10.1007/s11023-007-9062-6>
- Joyanes, L. (2009). La Computaci  n en Nube (Cloud Computing): El nuevo paradigma tecnol  gico para empresas y organizaciones en la Sociedad del Conocimiento. *Revista Icade. Revista de Las Facultades de Derecho y Ciencias Econ  micas y Empresariales*, 76, 95–111.  
<https://revistas.upcomillas.es/index.php/revistaicade/article/view/289>
- Joyanes, L. (2012). Cloud Computing. Notes for a spanish cloud computing strategy. Towards a national cloud computing strategy for administration and companies. *Journal of the Higher School of National Defense Studies*, 0(January-April), 83–104.
- Jula, A., Sundararajan, E., & Othman, Z. (2014). Cloud computing service composition: A systematic literature review. *Expert Systems with Applications*, 41(8), 3809–3824. <https://doi.org/10.1016/j.eswa.2013.12.017>
- Juzeleniene, S., Mikelioniene, J., Escudeiro, P., & Vaz de Carvalho, C. (2014). GABALL Project: Serious Games Based Language Learning. *Procedia - Social and Behavioral Sciences*, 136, 350–354.  
<https://doi.org/10.1016/j.sbspro.2014.05.340>
- Khribi, M. K., Jemni, M., & Nasraoui, O. (2008). Automatic recommendations for e-learning personalization based on Web usage mining techniques and information retrieval. *Eighth IEEE International Conference on Advanced Learning Technologies*, 241–245. <https://doi.org/10.1109/ICALT.2008.198>

- Killi, K., Moeller, K., & Ninaus, M. (2018). Evaluating the effectiveness of a game-based rational number training - In - game metrics as learning indicators. *Computers & Education*, 120, 13–28.  
<https://doi.org/10.1016/j.compedu.2018.01.012>
- Kitchenham, B. (2004). Procedures for performing systematic reviews. *University of Keele (Software Engineering Group, Department of Computer Science)*, 33(July), 1–28. <https://www.inf.ufsc.br/~aldo.vw/kitchenham.pdf>
- Kitchenham, B. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. Version 2.3. *University of Keele (Software Engineering Group, School of Computer Science and Mathematics) and Durham (Department of Computer Science)*, 57.  
<https://doi.org/10.1145/1134285.1134500>
- Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering - A systematic literature review. *Information and Software Technology*, 51(1), 7–15.  
<https://doi.org/10.1016/j.infsof.2008.09.009>
- Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering - A tertiary study. *Information and Software Technology*, 52(8), 792–805.  
<https://doi.org/10.1016/j.infsof.2010.03.006>
- Kroll, P., & Kruchten, P. (2003). *Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Addison Wesley.
- Kruchten, P. B. (1995). The 4+1 View Model of architecture. *IEEE Software*, 12(6), 42–50. <https://doi.org/10.1109/52.469759>
- Kwon, J., & Lee, Y. (2016). Serious games for the job training of persons with developmental disabilities. *Computers & Education*, 95(April), 328–339.  
<https://doi.org/10.1016/j.compedu.2016.02.001>
- Lafore, R. (2001). *Object-Oriented Programming in C++* (4th ed.). Sams.
- Lázaro-Cantabrina, J. L., Gisbert-Cervera, M., & Silva-Quiroz, J. E. (2018). Una rúbrica para evaluar la competencia digital del profesor universitario en el contexto latinoamericano. *Edutec. Revista Electrónica de Tecnología Educativa*, 63, 1–14. <https://doi.org/10.21556/edutec.2018.63.1091>
- Lee, K. (2017). *Foundations of Programming Languages* (I. Mackie (ed.); 2nd ed.). Springer. <https://doi.org/10.1007/978-3-319-70790-7>
- Leung, J., & Cockburn, A. (2021). Design Framework for Interactive Highlighting Techniques. *Design Framework for Interactive Highlighting Techniques*, 14(2–3), 1–176. <https://doi.org/10.1561/9781680839258>
- Lewis, W. (2017). *software testing and continuous quality improvement* (3rd ed.). Auerbach Publications. <https://doi.org/10.1201/9781439834367>
- López, C. (2016). El videojuego como herramienta educativa. Posibilidades y problemáticas acerca de los serious games. *Apertura. Revista de Innovación Educativa*, 8(1), 136–151.

## REFERENCIAS

- <https://dialnet.unirioja.es/servlet/articulo?codigo=5547036>
- López Caballero, V., Fragoso Díaz, O. G., Santaolaya Salgado, R., Rojas Pérez, J. C., & González Serna, J. G. (2019). Learning Web Services for E-learning in the Workplace. *IEEE Latin America Transactions*, 17(11), 1894–1901.  
<https://doi.org/10.1109/TLA.2019.8986429>
- Lozano, H. D. (2017). *Implementación de un juego serio multiplataforma para el desarrollo de la orientación espacial en niños de 6 a 8 años* [(Tesis de Licenciatura). Universidad Católica de Colombia].  
<https://repository.ucatolica.edu.co/bitstream/10983/15292/1/Documento - Trabajo de grado.pdf>
- Maggiorini, D., Ripamonti, L. A., Zanon, E., Bujari, A., & Palazzi, C. E. (2016). SMASH: A distributed game engine architecture. *IEEE Symposium on Computers and Communication (ISCC)*, 196–201.  
<https://doi.org/10.1109/ISCC.2016.7543739>
- Malaquias, R. F., Malaquias, F. F., & Hwang, Y. (2018). Understanding Technology Acceptance Features in Learning through a Serious Game. *Computers in Human Behavior*, 87(October), 395–402.  
<https://doi.org/10.1016/j.chb.2018.06.008>
- Marcano, B. (2008). Juegos Serios y Entrenamiento en la Sociedad Digital. *Revista Electrónica Teoría de La Educación. Educación y Cultura En La Sociedad de La Información*, 9(3), 93–107.  
<http://dialnet.unirioja.es/servlet/articulo?codigo=2778746&info=resumen>
- Marfisi-Schottman, I. (2012). *Méthodologie, modèles et outils pour la conception de Learning Games* [(Tesis doctoral). L'institut national des sciences appliquées de Lyon].  
<http://tel.archives-ouvertes.fr/tel-00762855/> %5Cn[http://liris.cnrs.fr/legadee/article/s/Marfisi\\_theseManuscrit\\_2012.pdf](http://liris.cnrs.fr/legadee/article/s/Marfisi_theseManuscrit_2012.pdf)
- Marfisi-Schottman, I., George, S., & Tarpin-Bernard, F. (2010). Tools and Methods for Efficiently Designing Serious Games. *4th European Conference on Games Based Learning, October*, 226–234. <http://free.iza.free.fr/articles/ECGBL-iza.pdf>
- Marsh, T. (2011). Serious games continuum: Between games for purpose and experiential environments for purpose. *Entertainment Computing*, 2(2), 61–68.  
<https://doi.org/10.1016/j.entcom.2010.12.004>
- Martínez, L. J. (2016). *Cómo buscar y usar información científica: Guía para estudiantes universitarios 2016*.  
[http://eprints.rclis.org/29934/7/Como\\_buscar\\_usar\\_informacion\\_2016.pdf](http://eprints.rclis.org/29934/7/Como_buscar_usar_informacion_2016.pdf)
- Massa, S. M. (2013). *Objetos de aprendizaje: metodología de desarrollo y evaluación de la calidad* [(Tesis doctoral). Universidad Nacional de La Plata (UNLP)]. <https://doi.org/10.35537/10915/26207>
- Massa, S. M., Evans, F., Bacino, G. A., Rico, C., Spinelli, A. T., Kühn, F. D., & Zapirain, E. A. (2017). *Videojuegos en Serio. Creando Serious Games para aprender jugando* (S. M. Massa & G. A. Bacino (eds.); 1st ed.). Universidad Nacional de Mar del Plata.

- [https://www.researchgate.net/publication/326156849\\_Elicitacion\\_de\\_Requerimientos\\_en\\_serious\\_game\\_un\\_caso\\_de\\_estudio](https://www.researchgate.net/publication/326156849_Elicitacion_de_Requerimientos_en_serious_game_un_caso_de_estudio)
- Massa, S. M., & Kühn, F. D. (2018a). Analíticas de Aprendizaje para Serious Games. *XX Workshop de Investigadores En Ciencias de La Computación (WICC2018)*, 1102–1105. <http://sedici.unlp.edu.ar/handle/10915/68396>
- Massa, S. M., & Kühn, F. D. (2018b). Learning Analytics in Serious Games: a systematic review of literature. *IEEE Biennial Congress of Argentina (ARGENCON)*, 1–5. <https://doi.org/10.1109/ARGENCON.2018.8646166>
- McMahon, M. (2009). Using the DODDEL model to teach serious game design to novice designers. In R. J. Atkinson & C. McBeath (Eds.), *Proceedings of Ascilite 2009* (pp. 646–653). The University of Auckland, Auckland University of Technology & Australasian Society for Computers in Learning in Tertiary Education. <http://www.ascilite.org/conferences/auckland09/procs/>
- Medvidovic, N., & Taylor, R. N. (2010). Software Architecture: Foundations , Theory , and Practice. *ACM/IEEE 32nd International Conference on Software Engineering*, 471–472. <https://doi.org/10.1145/1810295.1810435>
- Melegati, J., Goldman, A., Kon, F., & Wang, X. (2019). A model of requirements engineering in software startups. *Information and Software Technology*, 109, 92–107. <https://doi.org/10.1016/j.infsof.2019.02.001>
- Melgarejo-Torralba, M., Parras-Burgos, D., & Fernández-Pacheco, D. G. (2022). Hand-developed creative prototyping. Methodological proposal and experimentation. *Thinking Skills and Creativity*, 44, 101025. <https://doi.org/10.1016/j.tsc.2022.101025>
- Mell, P., & Grance, T. (2010). The NIST Definition of Cloud Computing. *Association for Computing Machinery. Communications of the ACM*, 53(6), 1–50.
- Meng, S., Pan, Z., Li, W., Xie, S., Liu, C., He, K., & Yang, H. (2010). The “4+1” view model on safe home system architecture. *Proceedings 2010 IEEE International Conference on Software Engineering and Service Sciences (ICSESS 2010)*, 352–355. <https://doi.org/10.1109/ICSESS.2010.5552450>
- Michael, D., & Chen, S. (2006). Serious Games: Games That Educate, Train, and Inform. In *Thomson Course Technology*. Thomson Course Technology. <http://portal.acm.org/citation.cfm?id=1051239>
- Ministerio de Educación de Chile. (2006). *Estándares en Tecnología de la Información y la Comunicación para la Formación Inicial Docente*. Ministerio de Educación de Chile.
- Mizutani, W. K., Daros, V. K., & Kon, F. (2021). Software architecture for digital game mechanics: A systematic literature review. *Entertainment Computing*, 38, 100421. <https://doi.org/10.1016/j.entcom.2021.100421>
- Mohr, F., Wever, M., & Hullermeier, E. (2018). On-the-Fly Service Construction with Prototypes. *IEEE International Conference on Services Computing*, 225–232. <https://doi.org/10.1109/SCC.2018.00036>
- Morales, V. G. (2013). Desarrollo de competencias digitales docentes en la

## REFERENCIAS

- educación básica. *Apertura. Revista de Innovación Educativa*, 5(1), 88–97. <http://www.udgvirtual.udg.mx/apertura/index.php/apertura/article/view/367/307>
- Moreno-Ger, P., Martínez-Ortiz, I., Sierra, J. L., & Fernández-Manjón, B. (2008). A Content-Centric Development Process Model. *Computer*, 41(3), 24–30. <https://doi.org/10.1109/MC.2008.73>
- Muñoz-Osuna, F. O., Medina-Rivilla, A., & Guillén-Lúgigo, M. (2016). Jerarquización de competencias genéricas basadas en las percepciones de docentes universitarios. *Educación Química*, 27(2), 126–132. <https://doi.org/10.1016/j.eq.2015.11.002>
- Muñoz, K., Kevitt, P. M., Lunney, T., Noguez, J., & Neri, L. (2011). An emotional student model for game-play adaptation. *Entertainment Computing*, 2(2), 133–141. <https://doi.org/10.1016/j.entcom.2010.12.006>
- Murazzo, M. A., Millán, I. F., Rodríguez, N. R., Segura, D., & Villafaña, D. A. (2010). Desarrollo De Aplicaciones Para Cloud Computing. *XVI Congreso Argentino de Ciencias de La Computación*, 941–949. <http://hdl.handle.net/10915/19374>
- Murazzo, M. A., Tinetti, F. G., Rodríguez, N. R., & Guevara, M. J. (2015). Infraestructura de Cloud Computing. *XVII Workshop de Investigadores En Ciencias de La Computación*, 1–4. <http://hdl.handle.net/10915/46197>
- Myers, G., Badgett, T., & Sandler, C. (2012). *The Art of Software Testing* (3rd ed.). John Wiley & Sons. <https://doi.org/10.1002/9781119202486>
- Nadolski, R. J., Hummel, H. G., van den Brink, H. J., Hoefakker, R. E., Slootmaker, A., Kurvers, H. J., & Storm, J. (2008). EMERGO: A methodology and toolkit for developing serious games in higher education. *Simulation & Gaming*, 39(3), 338–352. <https://doi.org/10.1177/1046878108319278>
- Nelson, M. J., Voithofer, R., & Cheng, S.-L. (2019). Mediating factors that influence the technology integration practices of teacher educators. *Computers & Education*, 128(January), 330–344. <https://doi.org/10.1016/j.compedu.2018.09.023>
- Nieto, E., & Bazán, P. A. (2013). *Diseño de aplicaciones SaaS sobre plataformas de Cloud Computing* [Universidad Nacional de La Plata (UNLP)]. <http://hdl.handle.net/10915/46834>
- Noll, J., Beecham, S., & Richardson, I. (2010). Global software development and collaboration: barriers and solutions. *ACM Inroads*, 1(3), 66–78. <https://doi.org/10.1145/1835428.1835445>
- Noll, J., Butterfield, A., Farrell, K., Mason, T., McGuire, M., & McKinley, R. (2014). GSD Sim: A Global Software Development Game. *IEEE International Conference on Global Software Engineering Workshops*, 15–20. <https://doi.org/10.1109/ICGSEW.2014.12>
- Nousiainen, T., Kangas, M., Rikala, J., & Vesisenaho, M. (2018). Teacher competencies in game-based pedagogy. *Teaching and Teacher Education*, 74, 85–97. <https://doi.org/10.1016/j.tate.2018.04.012>
- Ochoa, C., Villaizán, C., González de Dios, J., Hijano, F., & Málaga, S. (2016).

## REFERENCIAS

- Continuum, la plataforma de Formación Basada en Competencias. *Anales de Pediatría*, 84(4), 238.e1-238.e8. <https://doi.org/10.1016/j.anpedi.2015.12.002>
- Olayan, N., Patu, V., Matsuno, Y., & Yamamoto, S. (2013). A dependability assurance method based on data flow diagram (DFD). *2013 European Modelling Symposium*, 113–118. <https://doi.org/10.1109/EMS.2013.20>
- Ollsson, T., Toll, D., Wingkvist, A., & Ericsson, M. (2015). Evolution and Evaluation of the Model-View-Controller Architecture in Games. *4th International Workshop on Games and Software Engineering (GAS 2015)*, 8–14. <https://doi.org/10.1109/GAS.2015.10>
- OmniumGames. (2014, October). Spirits of Spring: Una emocionante aventura contra el bullying. *Game Journal*.  
<http://omniumgames.com/spirits-spring-una-emocionante-aventura-contra-el-bullying/>
- Padilla-Zea, N. (2011). *Metodología para el diseño de videojuegos educativos sobre una arquitectura para el análisis del aprendizaje colaborativo* [Universidad de Granada]. <http://hdl.handle.net/10481/19440>
- Padilla-Zea, N., Medina, N., Gutiérrez, F. L., Paderewski, P., López-Arcos, J. R., Núñez, M. P., & Rienda, J. (2015). Evaluación continua para aprendizaje basado en competencias: Una propuesta para videojuegos educativos. *Revista Iberoamericana de Informática Educativa*, 21(Enero-Junio), 25–38.  
<http://www.um.es/ead/red/42/UrbinaSalinas.pdf%5Cr%5Cn>
- Paternó, F. (1997). Formal reasoning about dialogue properties with automatic support. *Interacting with Computers*, 9(2), 173–196.  
[https://doi.org/10.1016/S0953-5438\(97\)00015-5](https://doi.org/10.1016/S0953-5438(97)00015-5)
- Pereira, A. B., Souza Jr, G. N., Monteiro, D. C., Barros, E. S., Costa, H. P., Nascimento, P. a, Marques, L. B., De Souza, D. G., Salgado, F. M., & Bessa, R. Q. (2012). A AIED Game to help children with learning disabilities in literacy in the Portuguese language. *Brazilian Symposium on Games and Digital Entertainment*, 134–143.  
[http://sbgames.org/sbgames2012/proceedings/papers/computacao/comp-full\\_17.pdf](http://sbgames.org/sbgames2012/proceedings/papers/computacao/comp-full_17.pdf)
- Pérez, B., & Salas, F. (2009). Hallazgos en investigación sobre el profesorado universitario y la integración de las TIC en la enseñanza. *Revista Electrónica Actualidades Investigativas En Educación*, 9(1), 1–25.  
<https://doi.org/10.15517/aie.v9i1.9381>
- Petcu, D., Iancu, B., Peculea, A., Dadarlat, V., & Cebuc, E. (2013). Integrating Cisco Packet Tracer with Moodle platform: Support for teaching and automatic evaluation. *RoEduNet International Conference 12th Edition: Networking in Education and Research*, 1–6. <https://doi.org/10.1109/RoEduNet.2013.6714190>
- Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64, 1–18. <https://doi.org/10.1016/j.infsof.2015.03.007>
- Plomp, T., Feteris, A., Pieters, J. M., & Tomic, W. (1992). *Ontwerpen van onderwijs*

- en trainingen []]. In T. Plomp, A. Feteris, J. M. Pieters, & W. Tomic (Eds.), *Designing education and training* (pp. 1–363).
- Pontual, T., Mendes de Andrade e Peres, F., Sales de Morais, D. C., & da Silva, G. (2018). Participatory methodologies to promote student engagement in the development of educational digital games. *Computers & Education*, 116, 161–175. <https://doi.org/10.1016/j.compedu.2017.09.006>
- Prasad, R., & Verma, G. (2016). *Software Engineering* (1st ed.). Khanna Book Publishing Co. (P) Ltd.
- Prendes, M. P. (2010). *Competencias TIC para la docencia en la universidad pública española: Indicadores y propuestas para la definición de buenas prácticas. Informe del proyecto financiado por la Secretaría de Estado de Universidades e Investigación del Ministerio de Educación*. [https://www.um.es/competenciastic/informe\\_final\\_competencias2010.pdf](https://www.um.es/competenciastic/informe_final_competencias2010.pdf)
- Prendes, M. P., Castañeda, L., & Gutiérrez, I. (2010). Competencias para el uso de TIC de los futuros maestros. *Comunicar. Revista Científica de Educomunicación*, 18(35), 175–181. <https://doi.org/10.3916/C35-2010-03-11>
- Prendes, M. P., & Gutiérrez, I. (2013). Competencias tecnológicas del profesorado en las universidades españolas. *Revista de Educación*, 361, 196–222. <https://doi.org/10.4438/1988-592X-RE-2011-361-140>
- Prendes, M. P., Gutiérrez, I., & Martínez, F. (2018). Competencia digital: una necesidad del profesorado universitario en el siglo XXI. *RED. Revista de Educación a Distancia*, 56, 1–22. <https://doi.org/10.1088/0022-3735/9/10/015>
- Qian, K., Xian, F., Lixin, T., Chong-Wie, X., & Díaz-Herrera, J. (2009). *Software Architecture and Design Illuminated* (1st ed.). Jones and Bartlett Publishers.
- Qian, M., & Clark, K. R. (2016). Game-based Learning and 21st century skills: A review of recent research. *Computers in Human Behavior*, 63(October), 50–58. <https://doi.org/10.1016/j.chb.2016.05.023>
- Queiruga, C., Fava, L., Gómez, S., Kimura, I. M., & Brown, M. (2012). El juego como estrategia didáctica para acercar la programación a la escuela secundaria. *XVI Workshop de Investigadores En Ciencias de La Computación (WICC)*, 358–362. <http://hdl.handle.net/10915/41365>
- Ralph, P., & Wand, Y. (2007). A Proposal for a Formal Definition of the Design Concept. In K. Lyytinen, P. Loucopoulos, J. Mylopoulos3, & B. Robinson (Eds.), *Design Requirements Engineering: A Ten-Year Perspective* (1st ed., pp. 103–136). Springer. [https://doi.org/10.1007/978-3-540-39962-9\\_55](https://doi.org/10.1007/978-3-540-39962-9_55)
- Rani, N. (2017). Review on Application of Pipe and Filter Architectural Style. *International Journal of Current Trends in Engineering & Research*, 2(9), 7–12. <https://pdfs.semanticscholar.org/3aae/fb33d4582089016f1fd12c9f66dd70f10e69.pdf>
- Robertson, J., & Nicholson, K. (2007). Adventure Author: A learning environment to support creative design. *Proceedings of the 6th International Conference on Interaction Design And Children (IDC 2007)*, 37–44.

<https://doi.org/10.1145/1297277.1297285>

Roblizo, M. J., & Cázar, R. (2015). Usos y competencias en TIC en los futuros maestros de educación infantil y primaria: hacia una alfabetización real para docentes. *Píxel-Bit. Revista de Medios y Educación*, 47, 23–39.  
<https://doi.org/10.12795/pixelbit.2015.i47.02>

Rodríguez-García, A.-M., Raso-Sánchez, F., & Ruiz-Palmero, J. (2019). Competencia digital, educación superior y formación del profesorado: un estudio de meta-análisis en la Web of Science. *Píxel-Bit. Revista de Medios y Educación*, 54, 65–82. <https://doi.org/10.12795/pixelbit.2019.i54.04>

Rodríguez, N. R., Murazzo, M. A., Villafaña, D. A., Valenzuela, A., Martín, A., & Chávez, S. (2014). Una propuesta para la incorporación de Cloud Computing en la currícula de Grado. *TE&ET. Revista Iberoamericana de Tecnología En Educación y Educación En Tecnología*, 12, 37–43.  
<http://teyet-revista.info.unlp.edu.ar/wp-content/uploads/2016/06/TEYET12-art04.pdf>

Rodríguez, N. R., Valenzuela, A., Villafaña, D. A., Murazzo, M. A., Chávez, S. B., & Martín, A. E. (2014). Una propuesta para la incorporación de Cloud Computing en la currícula de Grado. *TE&ET. Revista Iberoamericana de Tecnología En Educación y Educación En Tecnología*, 12, 37–43.  
<http://teyet-revista.info.unlp.edu.ar/wp-content/uploads/2016/06/TEYET12-art04.pdf>

Romero, M., & Turpo, O. (2012). Serious Games para el desarrollo de las competencias del siglo XXI. *RED. Revista de Educación a Distancia*, 34(1), 1–22. <https://revistas.um.es/red/article/view/233511>

Romero, M., Usart, M., & Ott, M. (2015). Can Serious Games Contribute to Developing and Sustaining 21st Century Skills? *Games and Culture*, 10(2), 148–177. <https://doi.org/10.1177/1555412014548919>

Rosenberg, J. M., & Koehler, M. J. (2015). Context and technological pedagogical content knowledge (TPACK): A systematic review. *JRTE. Journal of Research on Technology in Education*, 47(3), 186–210.  
<https://doi.org/10.1080/15391523.2015.1052663>

Ruiz-Requies, I., Rubia-Avi, B., Martínez-Rodríguez, R. A., & Fernández-Rodríguez, E. (2010). Formar al profesorado inicialmente en habilidades y competencias en TIC: perfiles de una experiencia colaborativa. *Revista de Educación*, 352(1), 49–78.  
[http://www.mecd.gob.es/revista-de-educacion/numeros-revista-educacion/numeros-anteriores/2010/re352/re352\\_07.html](http://www.mecd.gob.es/revista-de-educacion/numeros-revista-educacion/numeros-anteriores/2010/re352/re352_07.html)

Salas-Zárate, M., & Colombo-Mendoza, L. (2012). Cloud Computing: A review of PaaS, IaaS, SaaS Service and Providers. *Revista Lámpsakos*, 7(enero-junio), 47–57.

Sampiero, V. M., & Barragán, J. F. (2018). Análisis de la percepción de docentes, usuarios de una plataforma educativa a través de los modelos TPACK, SAMR y TAM3 en una institución de educación superior. *Apertura. Revista de Innovación Educativa*, 10(1), 116–131. <https://doi.org/10.18381/Ap.v10n1.1162>

- Sánchez, É. (2013). Tamagocours, un jeu numérique sur les règles qui encadrent les usages des ressources numériques. *VII Colloque «Questions de Pédagogies Dans l'enseignement Supérieur»*, 528–536.  
<http://www.colloque-pedagogie.org/node/433>
- Sánchez, É., & Emin-Martínez, V. (2014). Towards a Model of Play: An Empirical Study. *European Conference on Games Based Learning*, 2(October), 503–512.  
<https://search.proquest.com/docview/1674245678?accountid=15870>
- Sánchez, É., Emin-Martínez, V., & Mandran, N. (2015). Jeu-game, jeu-play, vers une modélisation du jeu. Une étude empirique à partir des traces numériques d'interaction du jeu Tamagocours. *Sciences et Technologies de L'Information et de La Communication Pour L'Éducation et La Formation*, 22, 9–45.  
[https://www.persee.fr/doc/stice\\_1764-7223\\_2015\\_num\\_22\\_1\\_1685](https://www.persee.fr/doc/stice_1764-7223_2015_num_22_1_1685)
- Sancho, J. M. (1998). La tecnología: un modo de transformar el mundo cargado de ambivalencia. In F. Segú (Ed.), *Para una tecnología educativa* (2nd ed.). Editorial Horsori.  
[https://www.researchgate.net/publication/48137361\\_Para\\_una\\_tecnologia\\_educativa](https://www.researchgate.net/publication/48137361_Para_una_tecnologia_educativa)
- Sandí-Delgado, J. C. (2019). Juegos serios para la indagación de competencias tecnológicas que puedan integrarse en la práctica pedagógica del profesorado. Una propuesta de aplicación en la Sede del Atlántico de la Universidad de Costa Rica (UCR). *TE&ET. Revista Iberoamericana de Tecnología En Educación y Educación En Tecnología*, 23, 103–105.  
<https://doi.org/10.24215/18509959.23.e13>
- Sandí-Delgado, J. C. (2020). Desarrollo de competencias digitales en el profesorado a través de juegos serios: Un estudio de caso aplicado en la Universidad de Costa Rica (UCR). *E-Ciencias de La Información*, 10(2), 1–27.  
<https://doi.org/10.15517/eci.v10i2.38946>
- Sandí-Delgado, J. C., & Bazán, P. A. (2017). *Cloud computing: posibilidades para la ejecución de juegos serios educativos as a service (JSEaaS)* [(Tesis de especialización). Universidad Nacional de La Plata (UNLP)].  
<http://hdl.handle.net/10915/63388>
- Sandí-Delgado, J. C., & Bazán, P. A. (2019). Educational Serious Games as a Service: Challenges and Solutions. *Journal of Computer Science & Technology*, 19(01), 66–80. <https://doi.org/10.24215/16666038.19.e07>
- Sandí-Delgado, J. C., & Bazán, P. A. (2020). A systematic literature review of methodologies used for the design of serious games . A comparative analysis. In M. Carmo (Ed.), *International Conference on Education and New Developments (END 2020)* (pp. 358–362). inScience Press.  
<https://doi.org/10.36315/2020end076>
- Sandí-Delgado, J. C., & Bazán, P. A. (2021). Diseño de juegos serios: Análisis de metodologías. *E-Ciencias de La Información*, 11(2), 1–24.  
<https://doi.org/10.15517/eci.v11i2.45505>
- Sandí-Delgado, J. C., & Cordero-Badilla, D. (2013). *La integración de las tecnologías de la información y la comunicación (TIC) en la docencia*

- universitaria: el caso de la Sede del Atlántico de la Universidad de Costa Rica.* (Tesis de Maestría). Sistema de Estudios de Postgrado (SEP), Universidad de Costa Rica (UCR).
- Sandí-Delgado, J. C., & Cruz-Alvarado, M. A. (2016). Propuesta metodológica de enseñanza y aprendizaje para innovar la educación superior. *Revista InterSedes*, 17(36), 1–38. <https://doi.org/10.15517/isucr.v17i36.27100>
- Sandí-Delgado, J. C., & Cruz-Alvarado, M. A. (2018). Análisis comparativo de juegos móviles educativos basados en posicionamiento. *Revista Intersedes*, 19(39), 146–170. <https://doi.org/10.15517/isucr.v19i39.34075>
- Sandí-Delgado, J. C., & Sanz, C. V. (2018a). *Análisis comparativo de juegos serios educativos. Indagación sobre sus posibilidades para la adquisición de competencias tecnológicas en la formación del profesorado* [(Tesis de especialización). Universidad Nacional de La Plata (UNLP)]. <http://hdl.handle.net/10915/65653>
- Sandí-Delgado, J. C., & Sanz, C. V. (2018b). Revisión y análisis sobre competencias tecnológicas esperadas en el profesorado en Iberoamérica. *Edutec. Revista Electrónica de Tecnología Educativa*, 66, 93–121. <https://doi.org/10.21556/edutec.2018.66.1225>
- Sandí-Delgado, J. C., & Sanz, C. V. (2020). Juegos serios para potenciar la adquisición de competencias digitales en la formación del profesorado. *Revista Educación*, 44(1), 1–18. <https://doi.org/10.15517/revedu.v44i1.37228>
- Sandí-Delgado, J. C., Sanz, C. V., & Lovos, E. N. (2018). *Juegos serios para la indagación de competencias tecnológicas que puedan integrarse en la práctica pedagógica del profesorado. Una propuesta de aplicación en la Sede del Atlántico de la Universidad de Costa Rica (UCR)* [(Tesis de MAestría). Universidad Nacional de La Plata (UNLP), Buenos Aires, Argentina]. <https://doi.org/10.35537/10915/71063>
- Sandí-Delgado, J. C., Sanz, C. V., & Lovos, E. N. (2022). Acceptance of Serious Games to Develop Digital Competencies in Higher Education. *The Electronic Journal of E-Learning (EJEL)*, 20(3), 351–367. <https://doi.org/10.34190/ejel.20.3.2181>
- Sanz, C. V., Artola, V., Bione, J., & Miceli, P. (2018). AstroCode in the Wild. *10th International Conference on Education and New Learning Technologies*, 7542–7548. <https://doi.org/10.1007/978-3-319-17542-3>
- Sanz, C. V., Nordio, M., & Artola, V. (2018). FraccionAR. Un juego para aprender sobre fracciones basado en Interacción Tangible. *XIII Congreso de Tecnología En Educación y Educación En Tecnología*, 84–92.
- Sauvé, L. (2009). Design tools for online educational games: Concepts and application. In Z. Pan, A. D. Cheok, & W. Müller (Eds.), *Transactions on Edutainment II: Vol. LNCS 5660* (pp. 187–202). Springer Berlín, Heidelberg. [https://doi.org/10.1007/978-3-642-03270-7\\_9](https://doi.org/10.1007/978-3-642-03270-7_9)
- Sawyer, B., & Smith, P. (2008). Serious games taxonomy. *Slides from the Serious Games Summit at the Game Developers Conference*, 1–54.

- <https://thedigitalentertainmentalliance.files.wordpress.com/2011/08/serious-games-taxonomy.pdf>
- Scacchi, W. (2017). Practices and Technologies in Computer Game Software Engineering. *IEEE Software*, 34(1), 110–116.  
<https://doi.org/10.1109/MS.2017.20>
- Scherer, R., Siddiq, F., & Teo, T. (2015). Becoming more specific: Measuring and modeling teachers' perceived usefulness of ICT in the context of teaching and learning. *Computers & Education*, 88(October), 202–214.  
<https://doi.org/10.1016/j.comedu.2015.05.005>
- Scherer, R., Tondeur, J., & Siddiq, F. (2017). On the quest for validity: Testing the factor structure and measurement invariance of the technology-dimensions in the Technological, Pedagogical, and Content Knowledge (TPACK) model. *Computers & Education*, 112(September), 1–17.  
<https://doi.org/10.1016/j.comedu.2017.04.012>
- Scherer, R., Tondeur, J., Siddiq, F., & Baran, E. (2018). The importance of attitudes toward technology for pre-service teachers' technological, pedagogical, and content knowledge: Comparing structural equation modeling approaches. *Computers in Human Behavior*, 80(March), 67–80.  
<https://doi.org/10.1016/j.chb.2017.11.003>
- Shojaei, E., Luque, E., Rexachs, D., & Epelde, F. (2018). Cloud computing application model for online recommendation through fuzzy logic system. *JCS&T. Journal of Computer Science & Technology*, 18(03), 267–274.  
<https://doi.org/10.24215/16666038.18.e30>
- Silva, J. E. (2012). Estándares TIC para la Formación Inicial Docente: una política pública en el contexto chileno. *Archivos Analíticos de Políticas Educativas*, 20(7), 1–15. <http://epaa.asu.edu/ojs/article/viewFile/962/963>
- Silva, J. E., Miranda, P., Gisbert, M., Morales, J., & Onetto, A. (2016). Indicadores para evaluar la competencia digital docente en la formación inicial en el contexto Chileno - Uruguayo. *RELATEC. Revista Latinoamericana de Tecnología Educativa*, 15(3), 55–69. <https://doi.org/10.17398/1695>
- Slivnik, B. (2022). Context-sensitive parsing for programming languages. *Journal of Computer Languages*, 73, 101172. <https://doi.org/10.1016/j.cola.2022.101172>
- Söbke, H., & Streicher, A. (2016). Serious Games Architectures and Engines. In R. Dörner, S. Göbel, M. Kickmeier-Rust, M. Masuch, & K. Zweig (Eds.), *Entertainment Computing and Serious Games. Lecture Notes in Computer Science* (Vol. 9970, pp. 1–26). Springer, Cham.  
<https://doi.org/10.1007/978-3-642-03503-6>
- Sommerville, I. (2016). *Software Engineering* (M. Borthakur (ed.); 10th ed.). Pearson Education.
- Soudani, N., Raggad, B. G., & Zouari, B. (2009). A formal design of secure information systems by using a formal secure Data Flow Diagram (FSDFD). *Post-Proceedings of the 4th International Conference on Risks and Security of Internet and Systems, CRiSIS 2009*, 131–134.

- <https://doi.org/10.1109/CRISIS.2009.5411965>
- Spinelli, A. T., & Massa, S. M. (2018a). Elicitación de Requerimientos, Centrada en el Usuario, para el Desarrollo de un Serious Game. *XIX Workshop de Investigadores En Ciencias de La Computación*, 1194–1198.  
<http://hdl.handle.net/10915/62882>
- Spinelli, A. T., & Massa, S. M. (2018b). Elicitación de Requerimientos Educativos en un Serious Game. *XIII Congreso de Tecnología En Educación y Educación En Tecnología*, 68–76. <http://sedici.unlp.edu.ar/handle/10915/68892>
- Spinelli, A. T., Massa, S. M., Evans, F., & Rico, C. (2017). Elicitación de Requerimientos en serious game, un caso de estudio. *III Workshop Sobre Creatividad e Innovación En Informática (III WCII)*, 1–12.  
[https://www.researchgate.net/publication/326156849\\_Elicitacion\\_de\\_Requerimientos\\_en\\_serious\\_game\\_un\\_caso\\_de\\_estudio](https://www.researchgate.net/publication/326156849_Elicitacion_de_Requerimientos_en_serious_game_un_caso_de_estudio)
- Spinelli, A. T., Massa, S. M., & Zapiain, E. A. (2016). La construcción narrativa de un Serious Game. *IV Congreso Internacional de Videojuegos y Educación*, 1–12.  
[https://www.researchgate.net/publication/326157041\\_La\\_construccion\\_narrativa\\_de\\_un\\_Serious\\_Game\\_The\\_narrative\\_construction\\_of\\_a\\_Serious\\_Game](https://www.researchgate.net/publication/326157041_La_construccion_narrativa_de_un_Serious_Game_The_narrative_construction_of_a_Serious_Game)
- Stavrev, S., Terzieva, T., & Golev, A. (2018). Concepts for Distributed Input Independent Architecture for Serious Games. *CBU International Conference Proceedings*, 6, 1166–1172. <https://doi.org/10.12955/cbup.v6.1310>
- Sung, H.-Y., & Hwang, G.-J. (2013). A collaborative game-based learning approach to improving students' learning performance in science courses. *Computers & Education*, 63, 43–51. <https://doi.org/10.1016/j.compedu.2012.11.019>
- Tapia, H. G. (2018). Actitud hacia las TIC y hacia su integración didáctica en la formación inicial docente. *Revista Electrónica Actualidades Investigativas En Educación*, 18(3), 1–29. <https://doi.org/10.15517/aie.v18i3.34437>
- Tejeda, R., & Sánchez del Toro, P. (2010). Estrategias de intervención para la formación de competencias profesionales en la educación superior. *Revista de Pedagogía Universitaria*, XV(5), 39–53.  
<http://cvi.mes.edu.cu/peduniv/index.php/peduniv/article/view/554/553>
- Teo, T. (2009). Modelling technology acceptance in education: A study of pre-service teachers. *Computers & Education*, 52(2), 302–312.  
<https://doi.org/10.1016/j.compedu.2008.08.006>
- Tondeur, J., Aesaert, K., Prestridge, S., & Consuegra, E. (2018). A multilevel analysis of what matters in the training of pre-service teacher's ICT competencies. *Computers & Education*, 122, 32–42.  
<https://doi.org/10.1016/j.compedu.2018.03.002>
- Tran, C. D., George, S., & Marfisi-schottman, I. (2010). EDoS: An authoring environment for serious games design based on three models. *4th European Conference on Games Based Learning, February 2015*, 393–402.  
[https://www.researchgate.net/publication/268379008\\_EDoS\\_An\\_authoring\\_environment\\_for\\_serious\\_games\\_design\\_based\\_on\\_three\\_models](https://www.researchgate.net/publication/268379008_EDoS_An_authoring_environment_for_serious_games_design_based_on_three_models)

- Tsai, M.-J., Huang, L.-J., Hou, H.-T., Hsu, C.-Y., & Chiou, G.-L. (2016). Visual behavior, flow and achievement in game-based learning. *Computers & Education*, 98(July), 115–129. <https://doi.org/10.1016/j.compedu.2016.03.011>
- Tseng, J.-J., Cheng, Y.-S., & Yeh, H.-N. (2019). How pre-service English teachers enact TPACK in the context of web-conferencing teaching: A design thinking approach. *Computers & Education*, 128, 171–182. <https://doi.org/10.1016/j.compedu.2018.09.022>
- UCR, U. de C. R. (2004). *Consejo Universitario, Acta de la Sesión N.º 4932* (pp. 1–30). Consejo Universitario, UCR.
- Unión Europea. (2006, December 30). Recomendación del Parlamento Europeo y del Consejo, de 18 de diciembre de 2006, sobre las competencias clave para el aprendizaje permanente (2006/962/CE). *Diario Oficial de La Unión Europea*, 394/10-394/18. <http://eur-lex.europa.eu/eli/reco/2006/962/oj>
- Vaca, C. (2012). *Paradigmas de Programación* (pp. 1–69). Departamento de informática.
- Van Der Vegt, W., Nyamsuren, E., & Westera, W. (2016). RAGE Reusable Game Software Components and Their Integration into Serious Game Engines. In G. M. Kapitsaki & E. Santana de Almeida (Eds.), *Software Reuse: Bridging with Social-Awareness. Lecture Notes in Computer Science* (Vol. 9679, pp. 165–180). Springer, Cham. [https://doi.org/10.1007/978-3-319-35122-3\\_25](https://doi.org/10.1007/978-3-319-35122-3_25)
- Van Der Vegt, W., Westera, W., Nyamsuren, E., Georgiev, A., & Martínez, I. (2016). RAGE Architecture for Reusable Serious Gaming Technology Components. *International Journal of Computer Games Technology*, 1–10. <https://doi.org/10.1155/2016/5680526>
- van Oijen, J., Vanhée, L., & Dignum, F. (2012). CIGA: A Middleware for Intelligent Agents in Virtual Environments. In M. Beer, C. Brom, F. Dignum, & V.-W. Soo (Eds.), *Agents for Educational Games and Simulations* (Vol. 7471, pp. 22–37). Springer, Heidelberg. <http://dl.acm.org/citation.cfm?id=2369028.2369031>
- Vega-Hernández, M.-C., Patino-Alonso, M.-C., & Galindo-Villardón, M.-P. (2018). Multivariate characterization of university students using the ICT for learning. *Computers & Education*, 121, 124–130. <https://doi.org/10.1016/j.compedu.2018.03.004>
- Velásquez, N., Estevez, E., & Pesado, P. M. (2018). Cloud Computing, Big Data and the Industry 4.0 Reference Architectures. *JCS&T. Journal of Computer Science & Technology*, 18(03), 258–266. <https://doi.org/10.24215/16666038.18.e29>
- Velurtas, E., & Bazán, P. A. (2018). Cloud Médicos - Privacidad , integridad y estandarización - Estado actual. *VI Jornadas de Cloud Computing & Big Data (JCC&BD)*, 29–35. <http://sedici.unlp.edu.ar/handle/10915/69677>
- Vinhas, L. (2016). *Fundamentals Of Structured Programming*.
- Vogel, J. O. (1970). Serious games: Clark C. Abt The Viking Press, \$5.95. *Business Horizons*, 13(3), 95. [10.1111/1467-954X.ep11200615%5Cnhttp://search.ebscohost.com/login.aspx?d](https://doi.org/10.1111/1467-954X.ep11200615%5Cnhttp://search.ebscohost.com/login.aspx?d=10.1111/1467-954X.ep11200615%5Cnhttp://search.ebscohost.com/login.aspx?d)

- irect=true&db=sih&AN=11200615&site=ehost-live
- Waag-Society. (2019). *Frequency 1550*. Mobile Learning Academy.  
<http://mobilelearningacademy.org/projects/frequency-1550/>
- Wrzesien, M., & Alcañiz, M. (2010). Learning in serious virtual worlds: Evaluation of learning effectiveness and appeal to students in the E-Junior project. *Computers & Education*, 55(1), 178–187. <https://doi.org/10.1016/j.compedu.2010.01.003>
- Yeh, Y.-F., Hsu, Y.-S., Wu, H.-K., & Chien, S.-P. (2017). Exploring the structure of TPACK with video-embedded and discipline-focused assessments. *Computers & Education*, 104(January), 49–64.  
<https://doi.org/10.1016/j.compedu.2016.10.006>
- Yessad, A., Labat, J.-M., & Kermorvant, F. (2010). SeGAE: A serious game authoring environment. *10th IEEE International Conference on Advanced Learning Technologies (ICALT 2010)*, 538–540. <https://doi.org/10.1109/ICALT.2010.153>
- Yu, C., & Prince, D. L. (2016). Aspiring School Administrators' Perceived Ability to Meet Technology Standards and Technological Needs for Professional Development. *JRTE. Journal of Research on Technology in Education*, 48(4), 239–257. <https://doi.org/10.1080/15391523.2016.1215168>
- Yulianto, B., Prabowo, H., Kosala, R., & Hapsara, M. (2017). Harmonik = ++(Web IDE). *Procedia Computer Science*, 116, 222–231.  
<https://doi.org/10.1016/j.procs.2017.10.044>
- Yun, Y.-S., Kim, S., Park, J., Kim, H., Jung, J., & Eun, S. (2020). Development of Neuromorphic Architecture Integrated Development Environment. *International Conference on Green and Human Information Technology*, 47–49.  
<https://doi.org/10.1109/ICGHIT49656.2020.00019>
- Yusoff, A., Crowder, R., & Gilbert, L. (2010). Validation of Serious Games Attributes Using the Technology Acceptance Model. *Second International Conference on Games and Virtual Worlds for Serious Applications*, 45–51.  
<https://doi.org/10.1109/VS-GAMES.2010.7>
- Zempoalteca, B., Barragán, J. F., González, J., & Guzmán, T. (2017). Formación en TIC y competencia digital en la docencia en instituciones públicas de educación superior. *Apertura. Revista de Innovación Educativa*, 9(1), 80–96.  
<http://www.udgvirtual.udg.mx/apertura/index.php/apertura/article/view/922>