Definiendo un modelo de proceso de *software* para la práctica del modding*

*Andrés Felipe Ceballos***
*Wilson Libardo Pantoja Yépez****
*Julio Ariel Hurtado*****

Recibido: 01/07/2019 • Aceptado: 27/11/2019
https://doi.org/10.22395/rium.v19n37a7

Resumen

El *modding* es una práctica de software difundida principalmente en la comunidad de videojuegos, en la que se involucran grupos interdisciplinares de *modders* con la finalidad de crear, mejorar y distribuir extensiones, llamadas *mods*, para *software* que ha sido liberado previamente. Lo anterior se realiza con la finalidad de ampliar las características de productos base, lo cual trae consigo nuevas aplicaciones, compatibilidad y extensibilidad, entre otras mejoras. Sin embargo, diferentes problemáticas, similares a las presentes en el proceso de desarrollo de *software*, se han visto involucradas dentro de esta práctica: barreras en la comunicación, ausencia de planeación, resultados inesperados y repetición de trabajo, entre otras. Así, en busca de integrar aspectos de la ingeniería de procesos para brindar soporte a estos grupos y disminuir los problemas presentes en este tipo de proyectos, se propone en el presente escrito el proceso de *software Kross Modding Process,* que brinda guía a los *modders* a través de información recogida en estudios previos sobre esta área.

Palabras clave: modding; proceso de *software*; ingeniería de *software*; modelo.

* Este trabajo forma parte del proyecto de investigación "Diseño de juegos pervasivos basados en experiencias de aprendizaje sensible al contexto" - Dispersa de las universidades del Cauca y de Granada, financiado parcialmente por el Ministerio de Economía y Competitividad de España y la Universidad del Cauca.

** Ingeniero de sistemas. Universidad del Cauca. Correo electrónico: aceballos@unicauca.edu.co. Orcid: https://orcid.org/0000-0002-5849-7924

*** Ingeniero de sistemas y magíster en computación. Profesor titular, Universidad del Cauca. Correo electrónico: wpantoja@unicauca.edu.co. Orcid: https://orcid.org/0000-0002-7235-6036.

**** Ingeniero en electrónica y telecomunicaciones, y doctor en ciencias de la Computación (Universidad de Chile). Profesor titular, Universidad del Cauca. Correo electrónico: jhurtado@dcc.uchile.cl. Orcid: https://orcid.org/0000-0002-2508-0962.

Defining a Software Process Model for Modding Practice

Abstract

Modding is a software practice, mainly spread in the videogame community, in which interdisciplinary groups of modders are involved with the aim of creating, improving and distributing extensions, called mods, for previously released software. This is done with the purpose of extending the characteristics of base products; thus, bringing new applications, compatibility, and extensibility, among other improvements. However, various problems, similar to those present in the software development process, have been involved in this practice: Communication barriers, lack of planning, unexpected results, and work repetition, among others. Thus, seeking to integrate aspects of process engineering to support these groups and reduce the problems present in this type of project, the software process "Kross Modding Process" is proposed, which provides guidance to modders through information collected in previous studies on this area.

Keywords: Modding; software process; software engineering; model.

Definindo um modelo de processo de *software* para a prática do modding

Resumo

O modding é uma prática de *software* difundida principalmente na comunidade de videogames, no qual se envolvem grupos interdisciplinares de *modders* com a finalidade de criar, melhorar e distribuir extensões chamadas *mods*, para software que foi lançado previamente. Isso se realiza com a finalidade de ampliar as características de produtos base, o qual traz consigo novas aplicações, compatibilidade e extensibilidade, entre outras melhoras. No entanto, diferentes problemáticas, similares às presentes no processo de desenvolvimento de *software*, viram-se envolvidas dentro dessa prática: barreiras na comunicação, ausência de planejamento, resultados inesperados e repetição de trabalho, entre outras. Desse modo, em busca de integrar aspectos da engenharia de processos para dar suporte a esses grupos e diminuir os problemas presentes nesse tipo de projetos, propõe-se no presente escrito o processo de software *Kross Modding Process*, que oferece guia aos *modders* por meio de informação recolhida em estudos prévios sobre essa área.

Palavras-chave: modding; processo de software; engenharia de *software*; modelo.

INTRODUCCIÓN

El *modding* es una práctica de grupos de desarrollo interdisciplinarios que buscan crear, mantener y distribuir extensiones respecto de un *software* desarrollado previamente, denominadas *mods*. A su vez, esta práctica se encuentra muy difundida en la comunidad de videojuegos: gracias a ella se obtienen productos derivados con nuevas características, tales como cambios en su comportamiento y estética. Con ello mejora la jugabilidad y se aportan nuevas historias, mecánicas y arte, entre otros aspectos [1] [2].

Mediante habilidades de exploración, transformación e innovación, los usuarios avanzados de los videojuegos establecieron esta práctica, originada como una ocupación de tiempo libre que buscaba ampliar la experiencia del producto original [3], impulsando la creación de nuevo contenido, basado en los gustos de los consumidores, mediante la experimentación y uso de herramientas dedicadas [4]. Así, esta práctica evolucionó con los usuarios y en colaboración con las casas desarrolladoras, que a su turno brindaron soporte, al tiempo que facilitaron herramientas de edición como *software development kits* (SDK) e incentivos para que los usuarios pudieran llevar a cabo sus proyectos, con lo cual se extendió la vida útil de sus productos [3] [5]. Gracias a esto se han conformado comunidades abiertas de *modders* [1], en las cuales se comparte el conocimiento, experiencia y recursos para la creación de mejores trabajos de *modding* de productos de *software*. Las webs Moddb y NexusMods, ejemplos de estas plataformas, están dedicadas a la gestión, desarrollo y distribución de *mods* [6].

El enfoque de las *mods* es conducido por el desarrollo del usuario final, también llamado *end-user development* [4], el cual permite que los creadores de contenido tomen el rol de desarrolladores, también llamados *fan-programadores* [7], para hacer uso de recursos propios como técnicas de texturizado y programación, hasta actividades de análisis como permisos, derechos de autor y requerimientos iniciales [8]. Sus actividades se realizan también de forma análoga a otros dominios de la construcción del *software* —planificación y pruebas, por ejemplo—. Así, de forma similar al desarrollo de *software*, el *modding* comparte algunas de las problemáticas comunes presentes en equipos de trabajo, como la falta de planeación; a la vez que ostenta otras más críticas, tales como actualizaciones inesperadas o daños en el núcleo del juego [1]. Sumado a lo anterior, y debido a las pocas prácticas de ingeniería en el *modding*, surgen aún más problemas propios de comunidades abiertas de trabajo, entre los cuales se encuentran los siguientes:

- Repetición del trabajo: surge en inconvenientes de coordinación entre los miembros de trabajo, y de la poca orientación hacia la definición y asignación de roles.

- Problemas de comunicación: aparecen junto a barreras como la cultura, el idioma, el horario y la geografía, entre otros.
- Resultados inesperados: una documentación inadecuada, o su ausencia, puede llevar a imprevistos, y a que su seguimiento resulte bastante complejo de realizar.

Estos problemas de comunicación y duplicación del trabajo hacen que el esfuerzo de los *modders* sea alto, a la vez que su avance vaya lento, puesto que se realizan esfuerzos adicionales en la integración de los módulos creados por cada miembro del equipo, además de posibles inconsistencias entre estos datos. Por otro lado, dada la naturaleza del *modding* como una actividad cultural y creativa [3] [9], su dinámica es conducida por prácticas empíricas, basadas en el conocimiento de los miembros del equipo de desarrollo. Sumado a lo anterior, debido al desconocimiento de productos y herramientas de soporte [10], se inician modificaciones por caminos errados que normalmente conducen a un mayor esfuerzo y, de modo eventual, a la inviabilidad de la modificación. Debido a estas razones, muchos proyectos de *modding* fracasan, con lo cual se pierden grandes esfuerzos en exploración, desarrollo y configuración [11].

Mediante la similitud de las actividades del *modding* respecto de las presentes en el desarrollo de *software,* y a través de la ingeniería de procesos de *software*, se propuso en este proyecto hacer uso de los datos recolectados a través de una revisión de la literatura científica y un estudio exploratorio previo con respecto al *modding*, para el diseño y posterior propuesta de un proceso *software* que brinde soporte a los grupos de *modders* en la realización de este tipo de proyectos.

1. MATERIALES Y MÉTODOS

1.1 Análisis desde la literatura

Algunos trabajos relacionados con respecto al *modding* identifican aspectos típicos de esta práctica: la similitud con la ingeniera de software, estudios de caso previos, guías para la realización de proyectos y la identificación de aspectos como roles y tipos de *mods*. Sin embargo, se observa la ausencia de un proceso definido que brinde soporte a la práctica del *modding* y que, a su vez, aporte a la comunidad tareas, artefactos y demás elementos necesarios para establecer un camino a la creación de *mods* de una forma productiva, al tiempo que reduzca el número de incidencias por posibles fallos dentro de su desarrollo. Así, tomando la información recolectada desde el estado del arte previo en *Entendiendo el modding como un proceso de ingeniería* [6], se logró abstraer algunos conceptos clave dentro de esta práctica, lo que permitió realizar una caracterización más precisa de la escena del *modding* en comunidades *gamer* [6]. Estos conceptos se presentan como actividades ejecutadas por los modders dentro

de sus proyectos creativos y se dividieron en 3 categorías: *pre-modding*, *modding* y *post-modding.*

1.1.1 Actividades de *pre-modding*

Análisis. Como una técnica basada en el desarrollo de *software* por parte del usuario final (*end-user development*) [4], los propietarios del *software* son los encargados de la generación de contenido nuevo al producto base, en forma de extensiones [12] [13]. Por lo tanto, en el *modding* se realizan actividades que constan de un análisis anterior a los cambios, con el objetivo de definir en un comienzo la meta de modificación y las herramientas y permisos disponibles, y organizar el equipo de trabajo. A continuación se identifican las actividades que componen el análisis:

- *Análisis de modificación:* dentro de esta actividad se realizan planeaciones de los cambios que se desea obtener, a la vez que se establecen las metas a conseguir en el desarrollo del proyecto. Estos análisis se evidencian en *mods* a videojuegos populares como StarCraft [11], Half Life II [14] y Minecraft [15]. En estas guías los *modders* plasman desde un comienzo lo que se desea realizar, a la vez que se definen los objetivos que se espera obtener con el proyecto.

- *Análisis del producto:* en este se incorpora una serie observaciones de los recursos disponibles en cuanto al *software* base, al igual que herramientas para realizar la labor de modificación. Se identifican los recursos necesarios y los que se encuentran disponibles para poner en marcha el proyecto [11], además de lo que el producto ofrece para ser modificado —arquitectura, plataforma, licencias y jugabilidad, entre otros— [14] [16] [17]. En la actualidad, las empresas desarrolladoras facilitan la labor de modificar partes de sus productos mientras no se realicen violaciones a su *Contrato de licencia para el usuario final* (CLUF) y, por lo general, brindan soporte a esta práctica. En consonancia con ello, es pertinente realizar una evaluación de las licencias del *software* que será objeto de uso en el proyecto de modificación [11].

Adquisición de herramientas especializadas. Tanto casas desarrolladoras, que apoyan la práctica de *modding* en sus productos, como *modders* experimentados que desarrollan herramientas de terceros (*third-party software*) [13], usan *software* dedicado a esta práctica. Estas herramientas permiten realizar la labor de exploración, edición e inclusión de archivos, y empaquetado de la extensión del *software* a modificar, con el fin de mejorar la eficiencia y facilitar la labor de algunas tareas que suelen ser complejas o repetitivas dentro de estos proyectos. Se encuentran dentro de estas herramientas kits de desarrollo de software (SDK) propios del *software* base, *software* de diseño asistido por computadora (CAD) y editores especializados de imagen, mapas, texto y sonido, entre otros [14] [16] [17].

Definición de roles. En tanto actividad colaborativa, el *modding* suele dar lugar a la conformación de equipos de personas con roles establecidos, que desempeñarán funciones específicas dentro del desarrollo del proyecto. Las responsabilidades y habilidades de cada integrante, así como su conocimiento, experiencia y talento, moldearán su respectivo rol (Tabla 1).

Los citados roles suelen asociarse a las motivaciones [18] de cada una de las personas pertenecientes al equipo, y además suelen variar en el transcurso del desarrollo del *mod* [7]. Los siguientes son algunos ejemplos de funciones que se pueden establecer en un proyecto de *modding*.

Tabla 1. Algunas funciones identificadas en proyectos de *modding*

Artista de texturas	Artista de sonido/música	*Beta tester*
Fundador	Artista conceptual	Proveedor de Servidor
Líder	Programador de herramientas	Administrador del servidor
Modelador	Escritores de ficción	*Mapper*
Programador	Animador	Artista gráfico
Webmaster	Diseñador web	*Skinner*
Scripter		

Fuentes: *Battlefield 1942* [7] y *Operation Flashpoint* [18]with special focus on the forms and consequences of collaboration between hobbyists. The case discussed in the article is the shooter-game Operation Flashpoint (OFP.

Se observa que la cantidad de funciones identificadas dentro de un grupo de trabajo puede llegar a ser extensa y muy variada, siempre en función de las capacidades de cada miembro.

1.1.2 Actividades de *modding*

Cambios en la jugabilidad. En los apartados del *modding*, la jugabilidad es un aspecto ampliamente modificado en estas comunidades [8] [9] [11]: reúne en conjunto las reglas, el funcionamiento, el diseño, la experiencia del videojuego con respecto al usuario final y con ella sus atributos —satisfacción, inmersión y emoción— [19].

Edición de scripts – código de alto nivel. Esta permitirá el cambio lógico del videojuego, con lo que se abrirá la posibilidad de inclusión de nuevos comportamientos en el sistema y la modificación de los existentes [17]. En este apartado, el *modder* podrá realizar cambios en las reglas o la física del juego, o bien habilitar contenido oculto.

Edición multimedia. Los recursos multimedia son elementos encargados de presentar la información al jugador en respuesta a sus interacciones. La edición de estos reúne todo lo correspondiente al manejo de gráficos, video, sonido y música, que van enlazados a los cambios que provee la extensión realizada [20].

Extensión de la vida del software base. Mediante la construcción y liberación de contenido derivado, las extensiones de *software* brindan a los usuarios finales contenido personalizado a través de la inmersión creativa: desde conversiones y edición multimedia hasta aspectos más avanzados, tales como corrección de errores y vulnerabilidades [1]. Este enfoque permite a las empresas realizar un seguimiento de sus productos en cuanto a los deseos de sus consumidores para futuros lanzamientos y mantenimiento del *software* liberado con anterioridad [13].

1.1.3 Actividades *post-modding*

Divulgación del mod. Dentro de la realización de un proyecto de modificación, los equipos de *modding* buscan distribuir el contenido realizado en la comunidad *gamer*. En ella, los equipos de trabajo buscan recibir retroalimentación frente a sus proyectos y, de igual manera, notificar posibles errores encontrados para un refinamiento posterior. En la actualidad existen comunidades en internet conformadas por millones de miembros que permiten alojar estos trabajos creativos y, posteriormente, compartirlos [13].

Mejoras posteriores. Posterior al resultado de la retroalimentación de la comunidad, y mediante el deseo de exploración, innovación y corrección [7] [18], se realizan mejoras a los trabajos realizados por los grupos de trabajo de *modding*. En ese sentido, se mejora la calidad del producto desarrollado y, con ello, se logra una mejor aceptación en la comunidad.

1.2 Datos desde el estudio exploratorio

Mediante la información recogida desde el estudio exploratorio realizado en [6] se pudieron obtener los siguientes datos, clasificados en tres secciones:

- Información con respecto a los *mods*
- Información con respecto a los *modders*
- Información con respecto a la administración del *modding*

Con respecto a los mods se encontró que un 38 % de los proyectos de *modding* de la comunidad de ModDB no han sido liberados, y que la mayoría de ellos lleva más de 6 años de desarrollo. Esto revela un porcentaje alto de pérdidas en ese sentido.

La cantidad de miembros dentro de las comunidades de *modding* analizadas supera 18 millones. Estos usuarios son los encargados de aportar a la escena del *modding* con el desarrollo de proyectos, en tanto que comparten conocimiento, habilidades y experiencia dentro de sus foros. En ese sentido, los participantes crean grupos de desarrollo y pequeños estudios independientes para la liberación continua de nuevos productos.

Las comunidades de ModDB y NexusMods proveen asistencia a los usuarios que desean empezar en proyectos de *modding*. Dentro de sus servicios brindan insumos básicos para el desarrollo, como foros de ayuda, seguimiento de actividades, manejo de versiones y herramientas de administración, entre otros.

1.3 El *modding* y los procesos de *software*

La ingeniería de procesos es un área perteneciente a la ingeniería de *software*, encargada de definir, implementar, medir y mejorar procesos de *software* [21]. Estos últimos se definen como conjuntos de actividades, participantes, estructuras, artefactos, herramientas y metodologías, entre otros elementos, cuyo fin es definir, desarrollar y mantener un producto de *software* [22]. Gracias al uso de procesos de *software* y de lenguajes de metamodelado como *Software Process Engineering Metamodel* (SPEM) se obtienen ventajas en el desarrollo, tales como:

- Facilitar la comprensión y comunicación humana.
- Facilitar la reutilización.
- Soporte para la gestión y mejora de procesos.
- Guiar la automatización de procesos.
- Dar soporte a la ejecución automática [21].

Lo anterior trae beneficios adicionales, como disponer de información para equipos de desarrollo cambiantes y facilitar la evaluación y certificación de estándares como CCMI e ISO, lo cual trae con ello ventajas comerciales y mejoras en la calidad del producto [6]. Debido a la gran diversidad de proyectos de desarrollo de *software* provistos por la complejidad, el contexto y el equipo de desarrollo, entre otros factores, el proceso de desarrollo de *software* no es universal. Sin embargo, algunas actividades son fundamentales y se encuentran presentes en la mayoría de los modelos de desarrollo [23]:

1) *Especificación de software*: se definen la funcionalidad y restricciones del software.
2) *Diseño e implementación:* se diseña el software de acuerdo con las especificaciones recogidas.
3) *Validación:* se realiza un proceso de esta naturaleza para asegurarse de que el producto cumpla con lo pactado.
4) *Evolución:* etapas de desarrollo y mantenimiento del *software* para que se adapte.

A través de la información presentada en el estado del arte y en el estudio exploratorio en [6] se identifican conceptos y prácticas que son similares entre los grupos de desarrollo de *modding* y el proceso de desarrollo de *software*. A continuación, se presenta una tabla comparativa entre estas dos prácticas.

Tabla 2: Comparación entre el *modding y el proceso de desarrollo de software*

Actividades del proceso de desarrollo de *software*	**Actividades en el *modding***
Especificación de *software*	Los equipos de *modders* establecen lo que se desea alcanzar con el proyecto de *modding*, así como las restricciones vigentes conforme a las licencias, permisos y recursos —humanos y de *software*— [11].
Diseño e implementación	Después de haberse establecido los objetivos en el inicio del proceso, los *modders* implementan los cambios en el sistema base a fin de desarrollar su producto [14] [16] [17].
Validación	Antes de la liberación de sus productos, los equipos de *modding* realizan la evaluación de las metas propuestas en un comienzo, para luego compartirlos en la web de la comunidad [3].
Evolución	Mediante la retroalimentación obtenida de los usuarios participantes en comunidades de *modding*, los equipos de desarrollo realizan correcciones a sus productos, así como mejoras y adaptaciones a otras plataformas [3] [24].

Fuente: [6].

Sumadas a las actividades fundamentales del proceso de desarrollo de *software*, se pudieron establecer otras similitudes con respecto al *modding* en [6], que se presentan a continuación:

- *Roles:* con el estudio exploratorio y el estado del arte se ha identificado que los miembros de los equipos de desarrollo de *modding* toman un rol dentro de sus proyectos y la comunidad con respecto a su talento, conocimiento y experiencia [7] [11] [18].

- *Actividades:* tanto en las guías presentes en el estado del arte [14] [16] [17] como en foros de las comunidades de *modding* se ha visto que los *modders* realizan pasos coherentes para desarrollar sus productos a partir del análisis, diseño, implementación y mantenimiento con la retroalimentación de los miembros de la comunidad.

- Artefactos: para la administración de los grupos de desarrollo de *modding,* así como para el establecimiento de tareas, se realiza el uso de artefactos de *software* mediante la documentación y la distribución de recursos multimedia dentro del equipo.

- Productos**:** dentro del estudio exploratorio de las comunidades de ambas webs se observa el lanzamiento constante de versiones del producto en función del cumplimiento de los objetivos trazados en el proyecto de *modding*.

- Equipos de desarrollo**:** mediante la colaboración de los miembros dentro de los equipos de *modding* se trabaja en conjunto para lograr los objetivos de desarrollo, de manera similar a las metodologías para el desarrollo de *software* [1].

Así, el *modding* abre la posibilidad de obtener, mediante la ingeniería de procesos, insumos para la mejora de esta práctica; ello involucra la definición de actividades, tareas y roles, y trae consigo la posibilidad de generar metodologías para el respaldo, de tal manera que se obtenga soporte y, con ello, mejores resultados en proyectos de *modding*.

1.4 Aspectos generales del *modding*

A través de la información recolectada desde el estado del arte y el estudio exploratorio realizado en [6], se procederá aquí a identificar aspectos comunes del *modding* como base para el posterior diseño del proceso de *software*.

1.4.1 Tipos de modding

Dentro del área del *modding* existen varios tipos de proyectos, los cuales se clasifican de acuerdo con su finalidad. En [1] se realiza una clasificación de 4 tipos de proyectos, a saber:

- *User interface:* los proyectos de este tipo se relacionan con las adaptaciones realizadas al *software* para mejorar la experiencia de los usuarios finales.

- *Game conversions:* es la forma más popular de *modding*. Se divide en dos partes, esto es, conversiones parcial y total (Tabla 1).

Tabla 1: Diferencias entre las conversiones parcial y total

Conversión parcial	Conversión total
Cambios en caracteres Cambios en los objetos Niveles, mapas, terrenos… Reglas del juego Mecánicas del juego	La conversión total incluye los cambios de la conversión parcial, pero en este caso se realiza una reconstrucción completa del juego, que dará como resultado un juego nuevo.

Fuente: elaboración propia.

- *Machinima:* generación de contenido multimedia a través del motor del videojuego que se modificará. Entre ellos se encuentran cambios en la historia del personaje, continuaciones o *spin-off (*proyecto que nace como extensión de otro anterior*)*. Actualmente existen grandes comunidades generadoras de este tipo de contenido, que generan *fan-fiction* (relatos de ficción, obras literarias o dramáticas escritas por

Fans de algún producto) extendiendo la experiencia en el juego con sus proyectos creativos.

- *Hacking closed systems:* este estilo de modificación consiste en el uso de técnicas como la ingeniería inversa para obtener información de partes cerradas del sistema —muchas veces con el fin de obtener ventaja frente a otros usuarios—. Las prácticas de este tipo suelen estar prohibidas por las casas desarrolladoras y la comunidad *gamer* en general, ya que se violan los acuerdos de la licencia del *software* y las normas de la comunidad misma.

1.4.2 Aspectos sociales del *modding*

En el trabajo de Sotamaa [18]with special focus on the forms and consequences of collaboration between hobbyists. The case discussed in the article is the shooter-game Operation Flashpoint (OFP se presenta un enfoque de la comunidad *modder* y algunos aspectos sociales en esta clase de grupos. La tabla 4 presenta una clasificación de las distintas motivaciones de los *modders* dentro del proyecto *Operation Flashpoint*.

Tabla 2: Motivaciones presentes en el *modding*

Jugabilidad	Disfrutar la experiencia del videojuego y con ello cubrir, mediante la construcción de contenido propio, la sensación de incompletitud del producto.
Hacking	Descubrir el funcionamiento interno del producto.
Investigación	Dada la naturaleza del videojuego (*Operation Flashpoint*), se realiza una observación desde el punto militar.
Expresión artística	La experiencia de la creación en el apartado del arte.
Cooperación	Creación de ideas con otros miembros del equipo y el entusiasmo por trabajar hacia un objetivo compartido.

Fuente: elaboración propia.

Estas motivaciones van de la mano con el análisis de Postigo [7] respecto a los temas que motivan a los *modders* y, en especial, a los *mappers*. A partir de sus creaciones es posible realizar una clasificación de tres temas centrales que los motivan: el primero, su deseo de hacer una contribución artística a las comunidades y obtener retroalimentación de ellas; el segundo, la identificación que sienten con su creación y su apropiación al producto mediante el diseño de elementos únicos que les representan algún significado fuerte; y el tercero, el efecto de su acción en el campo laboral, por cuanto cada proyecto se suma a su lista de experiencia y mejora sus oportunidades de conseguir un trabajo lucrativo en la industria del desarrollo de videojuegos.

1.4.3 Licencias híbridas

En el trabajo de Scacchi [1] se muestra un enfoque de licencias híbridas en que se parte de un *software* propietario, pero con modificaciones abiertas (Figura 1). Mediante este enfoque de trabajo se permite reutilizar recursos producidos en la

comunidad de *modding*, cuestión que posibilitaría la retroalimentación y mejora en los productos que se desarrollan —siempre que se mantenga respeto por el núcleo o *software* base (privativo)—.

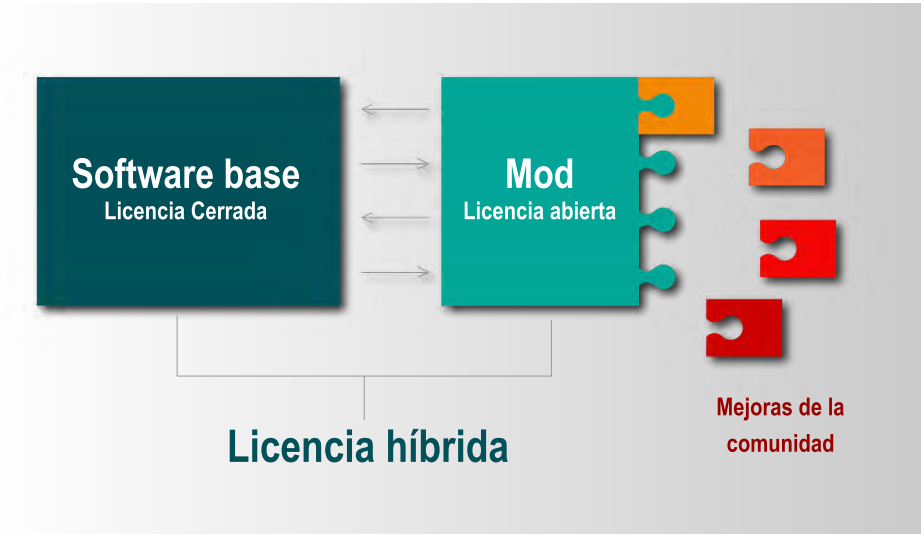

Figura 1. Enfoque de licencias híbridas

Fuente: elaboración propia.

1.5 Metodología de Formalización

Para lograr la formalización y diseño de "*Kross Modding Process*", como un proceso que permita apoyar la práctica del modding mediante la colaboración, se procedió a realizar las actividades propuestas en el metaproceso definido por Ruiz [22], que se divide en tres fases:

1) *Planeación:* según lo descrito en el metaproceso de Ruiz, esta fase consiste en entender las necesidades de la empresa para la cual se diseñarán o mejorarán los procesos de software. Esta etapa involucra, a su turno, tres tareas principales (figura 2):

   a. *Analizar el contexto de la empresa:* radica en conocer las metas de negocio de empresa y el modo en que se alcanzan.

   b. *Definir y planear actividades:* se reúnen representantes de cada área, interesados y usuarios finales con fin de mostrar las ventajas de disponer un proceso definido en la empresa, establecer su alcance, identificar las principales actividades y designar a personal de cada área para funcionar como enlace de comunicación con los ingenieros de procesos.

c. *Entrenar a los stakeholders:* tras conocerse las metas y actividades, se realiza una familiarización con el proceso, el lenguaje y la herramienta de modelado para efectos de mantenimiento y evolución posteriores, mediante la participación activa de los miembros.

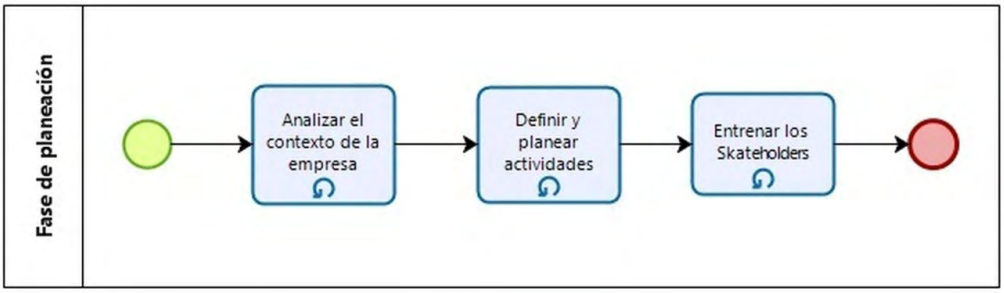

Figura 2. Tareas que componen la fase de planeación

Fuente: metaproceso de Ruiz [22].

2) *Ejecución:* esta fase del metaproceso consta de cinco tareas que permiten obtener la información de los procesos de *software* de la empresa, que luego darán pie a analizar, diseñar y especificar un prototipo mediante el lenguaje SPEM para realizar, finalmente, una validación con los interesados del proyecto, lo cual involucrará refinar u optimizar el producto final (figura 3).

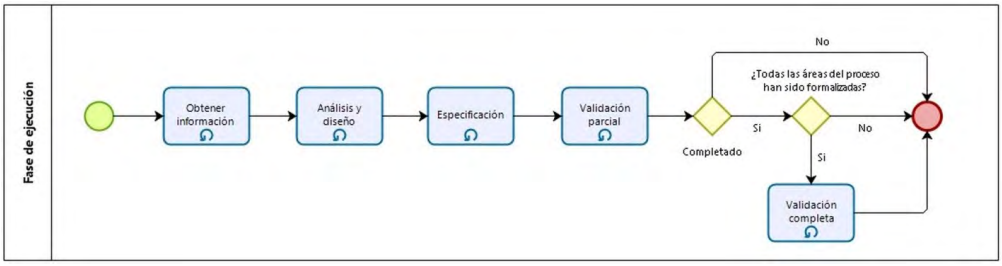

Figura 3. Fase de ejecución

Fuente: metaproceso de Ruiz [22].

3) *Entrega:* esta etapa consta de una sola tarea, llamada *despliegue del proceso.* Consiste en entregar a la empresa desarrolladora el proceso mediante un lenguaje como SPEM, desplegado en sus computadoras, en aras de tenerlo como referencia e información para los miembros para que evolucione y se adapte a las necesidades de la empresa (figura 4).

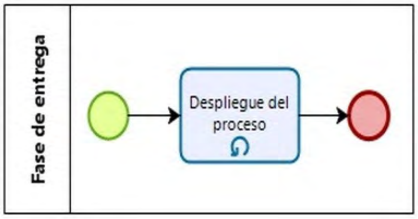

Figura 4. Tarea de despliegue

Fuente: metaproceso de Ruiz [22].

2. RESULTADOS

Mediante los datos recogidos en la literatura científica y el estudio exploratorio previo, sumado a la revisión de aspectos generales del *modding,* y a través de la metodología de formalización de Ruiz [22], se diseñó un proceso de *software* denominado *Kross Modding Process*, que se presenta a continuación.

2.1 Kross Modding Process

Este es un proceso de *software* para el desarrollo de la práctica *modding*. Está diseñado para ofrecer soporte a sus principales actividades de forma guiada, mediante tareas que abarcan la administración del proyecto, así como construcción y mejoras a través de la coordinación y colaboración de los miembros del equipo de trabajo (Figura 5).

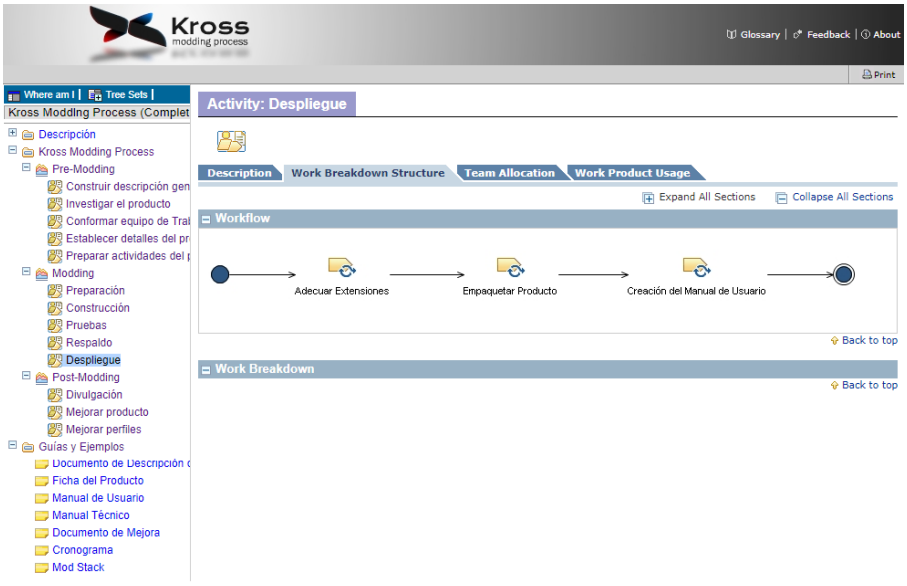

Figura 5. Versión web del proceso

Fuente: elaboración propia.

El proceso busca solucionar algunas problemáticas detectadas en la comunidad y la escena del *modding* —tales como *retrabajo*, problemas de comunicación y ausencia de definición de tareas y roles— mediante el uso de pasos coherentes para principiantes, la comunicación continua y la clasificación de la información obtenida en el proyecto.

2.1.1 Principios

Kross Modding Process se basa en cinco principios fundamentales, que buscan guiar a los participantes del proyecto mediante las buenas prácticas internas en un equipo de desarrollo (Tabla 3).

Tabla 3. Principios de *Kross Modding Process*

Principio	Descripción
1. Respeto	En la práctica del *modding* es importante considerar el respeto hacia el trabajo ajeno. Ello implica respetar dos elementos puntuales: de un lado, las licencias y las restricciones en los productos que se van a modificar, para evitar con ello el uso de herramientas que violen la propiedad intelectual de los recursos involucrados (incluido el *software*); y de otro, el trabajo de otros ***modders*** dentro de la comunidad y conforme a ello realizar la respectiva acreditación cuando se usen contenidos ajenos.
2. Multidisciplinariedad	"Solo trabajarás más rápido, pero acompañado llegarás más lejos". Las múltiples habilidades de los miembros de un equipo promueven la creación de buen contenido. Por lo tanto, es necesario contar con estos talentos y capacidades en la integración del material para lograr mejores resultados y, con ello, mejores *mods*.
3. Colaboración	La colaboración es necesaria dentro de la escena del *modding*: se requiere en este sentido el concierto de los miembros del equipo de desarrollo, así como trabajo, herramientas, contenido y opinión de terceros.
4. Modularidad	La base de un proyecto de *modding* es crear módulos cuya naturaleza es de extensión conectable al producto de *software* que será objeto de modificación, y requiere su previa adquisición para funcionar. Un *mod* no funciona como un producto aislado.
5. Extensibilidad	La naturaleza de la escena del *modding* es creativa y abierta. Se busca que los productos de *software* se expandan en características y en mejoras.

Fuente: elaboración propia.

2.1.2 Fases

Kross Modding Process consta de tres fases, que representan los estadios de un proyecto *modding* (Figura 6). A su vez, estas fases se dividen en trece actividades que parten de la investigación del producto base hasta la mejora del producto *mod* final.

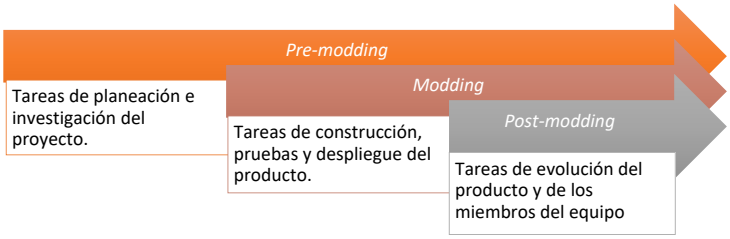

Figura 6. Fases de *Kross Modding Process*

Fuente: elaboración propia.

2.1.3 Roles

Mediante la abstracción de algunas funciones comunes en proyectos de *modding,* y a partir de la literatura y el estudio exploratorio, se identificaron tres roles principales, cuyas actividades y tareas se disponen en función de la capacidad del participante y el tipo de proyecto en desarrollo. A continuación, se describen los roles presentes en el proceso (Figura 7).

- *Líder del equipo:* se encarga de dirigir las actividades del equipo para conseguir buenos resultados en el desarrollo del proyecto. Además, establece roles y responsabilidades, facilita documentación y herramientas a sus miembros, fija metodologías para el desarrollo, y socializa conceptos sobre la idea de modificación.
- *Modder:* encargado de dar lugar a los cambios en el software base. Este miembro puede desempeñar uno o varios tipos de funciones, según sus habilidades y talentos (tabla 6).
- *Participante:* desempeña funciones alternas a las tareas de modificación dentro del grupo de trabajo. Con respecto a su talento y capacidad para aportar en el desarrollo del producto final, las funciones que puede desempeñar un participante pueden relacionarse con tareas como generador de retroalimentación, divulgador en redes sociales y *webmaster, entre otras.*

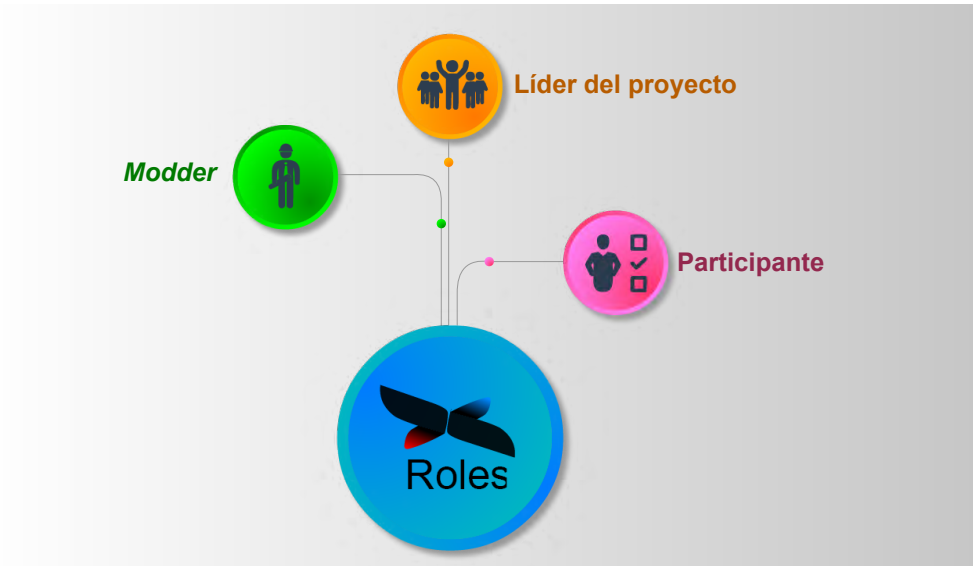

Figura 7. Roles de *Kross Modding Process*

Fuente: elaboración propia.

Sumado a lo anterior, cada miembro del equipo de desarrollo puede desempeñar múltiples funciones con respecto a su rol; en la Tabla 4 se muestran algunos ejemplos de ellas en un proyecto típico de *modding*.

Tabla 4: Algunas funciones presentes en los roles del proceso

Rol \ Clasificación	Administración	Arte	Conceptos	Programación
Líder del proyecto	• Director • Administrador de repositorios • Facilitador		• Analista • Planificador	
Modder		• Diseñador de interfaces • *Mapper* • Texturizador • Artista de sonido • Modelador 3D/2D • Animador	• Creador de personajes • Diseñador de niveles	• Programador de herramientas • *Scripter* • Editor de características • Programador general
Participante	• Generador de retroalimentación • *Webmaster* • Divulgador en redes	• Artista conceptual	• Historiador	

Fuente: elaboración propia.

2.1.4 Artefactos

Kross Modding Process ofrece diez artefactos que brindan la documentación necesaria para el desarrollo del proyecto de *modding* (figura 8), cuya finalidad es guiar a los miembros del equipo para facilitar la colaboración; se describen a continuación.

- *Cronograma del proyecto:* se compone de actividades que se delimitan por duración y tienen fechas de inicio y fin. Este artefacto permite al grupo de desarrollo gestionar las actividades a cumplir y, con ello, administrar el tiempo y recursos para lograrlo.
- *Documento de descripción del proyecto:* recoge los datos del diseño del proyecto. En él se incluyen las ideas principal y secundarias; la información del producto que será objeto de modding; los datos de los miembros del equipo y sus responsabilidades; y los objetivos, enlaces de las herramientas y recursos en línea.
- *Ficha del producto:* en este documento se recoge la información obtenida sobre los criterios para la elección, licencia, permisos y prestaciones para el proyecto que posee el producto base.
- *Mod stack:* este artefacto agrupa los modding items según su importancia en el desarrollo del proyecto, para después priorizarlos. Queda a disposición del equipo definir los criterios de priorización.

- *Manual técnico del mod:* se incluyen aquí todos los datos correspondientes a las modificaciones que hayan resultado de las tareas de modding del proyecto. Esta información funciona como referencia para el equipo de modders, así como para pruebas técnicas del producto, corrección de errores y posible adaptación de nuevas modding tasks.
- *Incremento:* representa cada liberación previa del mod, la cual se traduce a un conjunto de modding tasks completadas en la fase de modding.
- *Mod:* es el resultado del proyecto de modding que está en ejecución. Este es una extensión de software que se conectará al software base del proyecto para funcionar. Cada vez que se completa la fase de modding, el resultado es la liberación de una nueva versión del mod (incremento).
- *Manual de usuario:* documento en el que se consolida la información necesaria para la instalación y el manejo del mod. Dentro de él se describen al usuario los pasos a seguir para poner a marchar el mod.
- *Parche:* el propósito de un parche de software es corregir errores imprevistos en el mod en tiempo de liberación que impidan su ejecución. Estos parches vienen en forma de archivos, ejecutables o comprimidos que se liberan al usuario final con su respectiva descripción.
- *Documento de mejora:* tiene como finalidad recopilar las ideas de mejora para el mod que puedan surgir gracias a la retroalimentación de los usuarios finales. Esto con el fin de ampliar la experiencia del producto, de tal suerte que sus características mejoren en desarrollos futuros para aumentar su calidad.

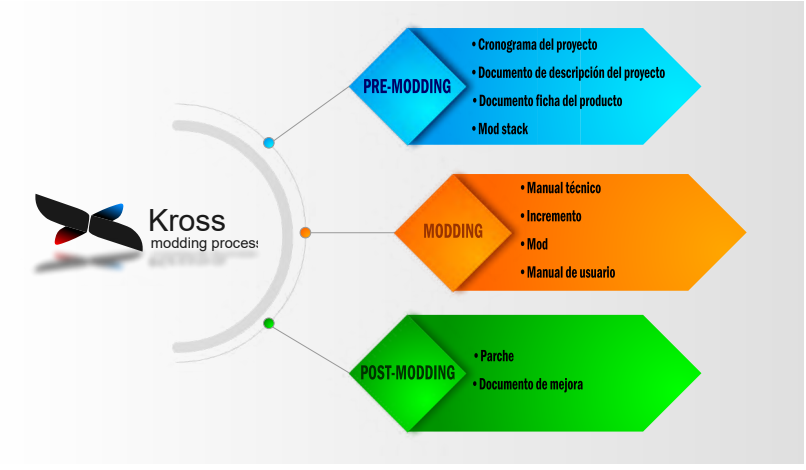

Figura 8. Artefactos de *Kross Modding Process*

Fuente: elaboración propia.

3. DISCUSIÓN

De acuerdo con el planteamiento del proyecto, y a través de la información recogida desde el estado del arte y el estudio exploratorio previo, *Kross Modding Process* provee una base para que grupos de desarrollo de *modding* puedan comenzar proyectos de esta naturaleza con actividades, tareas, roles e información definidos. Frente a los problemas identificados mediante esta investigación, *Kross Modding Process* permite reducir inconvenientes comunes dentro de los proyectos de *modding* como el retrabajo, problemas de comunicación y resultados inesperados a través de la documentación y definición de actividades y roles, a la par con el uso de los artefactos de soporte.

Debido a la naturaleza del *modding* y la heterogeneidad de sus proyectos, *Kross Modding Process* busca evolucionar y adaptarse a sus usuarios mediante el refinamiento, dado por experiencias reales en proyectos de *modding*. De igual manera, el proceso busca abordar un tema aún desconocido en la ingeniería de procesos, con lo que constituye un proyecto de investigación preliminar que puede conducir a futuras investigaciones en este campo (tales como la definición de metodologías enfocadas a la extensión de *software*).

4. CONCLUSIONES

El *modding* es una práctica en crecimiento que trae consigo ventajas y oportunidades de negocio para empresas productoras y estudios independientes de videojuegos, así como para los usuarios finales que actúan como desarrolladores: mejoran sus perfiles al tiempo que aportan a la innovación de nuevos productos en el mercado. Sin embargo, el *modding* ha sido poco abordado en el área de la ingeniería del *software* dado que su naturaleza es más creativa y empírica, y sus proyectos son multidisciplinares. En ese sentido el *Kross Modding Process* propuesto en este trabajo busca servir de guía para los *modders* y dar solución a problemas típicos en ese tipo de proyectos.

Este proceso es el resultado de seguir una metodología investigativa empírica. Aun cuando se encuentra en mora de aplicarse en diferentes escenarios, brinda las bases iniciales para sistematizar el *modding* como un proceso de desarrollo de *mods*. Es importante aclarar que el autor principal de esta investigación es un *modder* que ha abordado esta temática durante varios años y conoce cómo funcionan las comunidades, su estilo de trabajo y los productos que desarrollan, así como las limitaciones y dificultades de esta práctica.

Como proyecto a futuro se busca realizar una evaluación, mediante varios estudios de caso, encaminada a medir aspectos como la calidad y la productividad, e identificar limitaciones, con la finalidad de refinar el proceso adaptándolo a las diferentes

necesidades de grupos independientes de *modders* y a los distintos ámbitos de trabajo por los cuales se caracterizan los proyectos de esta naturaleza.

REFERENCIAS

[1] W. Scacchi, "Modding as an Open Source Approach to Extending Computer Game Systems", *Int. J. Open Source Softw. Process.*, vol. 3, n.º 3, pp. 36-47, 2011. DOI: 10.4018/jossp.2011070103.

[2] J. Kücklich, "Precarious Playbour : Modders and the Digital Games Industry The History of Modding The Economy of Modding", *The Fibreculture Journal*, n.° 5, pp. 1-9, 2005.

[3] M. Trenta, "La gestión de las comunidades de modding entre explotación y participación" en *Actas – V Congreso Internacional Latino de Comunicación*, pp. 1-15, 2013.

[4] H. Lieberman, F. Paternò y V. Wulf, "End User Development: An Emerging Paradigm", *End User Dev.*, vol. 9, pp. 9-16, 2006. DOI: 10.1007/1-4020-5386-X.

[5] S. Guthals, S. Foster, y L. Handley, *Minecraft modding for kids for dummies*, Hoboken: John Wiley & Sons, 2015.

[6] A. Ceballos, L. Pantoja y J. A. Hurtado, "Entendiendo el Modding como un Proceso de Ingeniería", en *Ciencias Computacionales*, Cartagena, pp. 175-186, 2018.

[7] H. Postigo, "Of Mods and Modders: Chasing Down the Value of Fan-Based Digital Game Modifications", *Games Cult.*, vol. 2, n.o 4, pp. 300-313, 2007. DOI: 10.1177/1555412007307955.

[8] W. Scacchi, "Modding as a basis for developing game systems", en *Proceeding of the 1st international workshop on Games and software engineering - GAS '11*, p. 5, 2011. DOI: 10.1145/1984674.1984677.

[9] A. Unger, "Modding as Part of Game Culture", en *Computer Games and New Media Cultures*, Dordrecht: Springer Netherlands, 2012, pp. 509-523. DOI: 10.1007/978-94-007-2777-9_32.

[10] G. Poderi y D. J. Hakken, "Modding a free and open source software video game: "Play testing is hard work", *Transformative Works and Cultures*, vol. 15, 2013. DOI: 10.3983/twc.2014.0493.

[11] D. Johnson, "StarCraft fan craft: Game mods, ownership, and totally incomplete conversions", *Velv. Light Trap*, vol. 64, n.o 1, pp. 50-63, 2009. DOI: 10.1353/vlt.0.0041.

[12] O. Sotamaa, "Computer game modding, intermediality and participatory culture", *New Media*, pp. 1-26, 2003.

[13] K. A. Moody, "Modders: Changing the game through user-generated content and online communities.", *Diss. Abstr. Int. Sect. A Humanit. Soc. Sci.*, vol. 75, n.° 11-A(E), 2015. DOI: 10.17077/etd.5ak8cz3w.

[14] E. Guilfoyle, *Half-Life 2 mods for dummies*, Hoboken: John Wiley & Sons, 2007.

[15] A. Gupta y A. Gupta, *Minecraft modding with Forge*, Estados Unidos: O'Reilly Media, 2015.

[16] E. Guilfoyle, *Quake 4 mods for dummies*, Hoboken: John Wiley & Sons, 2006.

[17] J. Van Gumster y C. Ammann, *Farming simulator modding for dummies*, Hoboken: John Wiley & Sons, 2014.

[18] O. Sotamaa, "When the Game Is Not Enough: Motivations and Practices Among Computer Game Modding Culture", *Games Cult.*, vol. 5, n.° 3, pp. 239-255, 2010, https://doi.org/10.1177/1555412009359765.

[19] J. L. G. Sánchez, N. P. Zea y F. L. Gutiérrez, "Playability: How to identify the player experience in a video game", 2009, https://doi.org/10.1007/978-3-642-03655-2_39.

[20] R. Hong, "Game Modding, Prosumerism and Neoliberal Labor Practices", *Int. J. Commun.*, vol. 7, pp. 984-1002, 2013.

[21] F. Ruiz y J. Verdugo, "Guía de Uso de SPEM 2 con EPF Composer", *Composer*, vol. 3, p. 93, 2008. DOI: 10.13140/2.1.1455.9049.

[22] P. Ruiz, A. Quispe, M. C. Bastarrica y J. A. H. Alegría, "Formalizing the Software Process in Small Companies" 2012.

[23] P. Letelier, "Proceso de Desarrollo de Software", *Univ. Politécnica Val.*, pp. 1-14, 2003.

[24] T. Sihvonen, *Players Unleashed! Modding the Sims and the Culture of Gaming*, Amsterdam: Amsterdam University Press, 2011. DOI: 10.5117/9789089642011.