

# **Un modelo para la solución de requerimientos no alineados: El caso del Software lúdico para la divulgación**

**Héctor G. Pérez-González**  
Universidad Autónoma de San Luís Potosí  
[hectorgerardo@acm.org](mailto:hectorgerardo@acm.org)

**Rosa M. Martínez-García**  
Universidad Autónoma de San Luís Potosí  
[rosma.fi.uaslp@gmail.com](mailto:rosma.fi.uaslp@gmail.com)

**Francisco E. Martínez-Pérez**  
Universidad Autónoma de San Luís Potosí  
[fcoemtz@gmail.com](mailto:fcoemtz@gmail.com)

**Sandra Nava-Muñoz**  
Universidad Autónoma de San Luís Potosí  
[senavam@uaslp.mx](mailto:senavam@uaslp.mx)

**Alberto S. Nuñez-Varela**  
Universidad Autónoma de San Luís Potosí  
[alberto\\_snv@hotmail.com](mailto:alberto_snv@hotmail.com)

**Resumen:** Este artículo propone lineamientos metodológicos para ser aplicados en el proceso de requerimientos cuando estos se presentan como “No alineados”. Se considera que esta situación se da cuando los objetivos del propietario del software son distintos a los del usuario del mismo. Como caso de estudio se utiliza el desarrollo de software lúdico (también conocido como juegos digitales, videojuegos, o juegos serios) con fines de divulgación. Esto involucra generalmente un conjunto de stakeholders más heterogéneo que el del software convencional. Adicionalmente, si el software se utiliza con objetivos de divulgación de ciencia, tecnología o innovación se agrega la complicación de la presencia de objetivos no alineados ya que el usuario final puede ignorar el objetivo real del software que utilizará. Consecuentemente el análisis de requerimientos cobra una importancia crucial. La literatura presenta propuestas para adaptar el proceso completo de software con objetivos educativos, sin poner atención al proceso de requerimientos ni a los objetivos de divulgación. Esta propuesta puede ser extrapolada a cualquier proceso de Ingeniería de requerimientos de Software donde se involucren requerimientos no alineados.

**Palabras Clave:** Ingeniería de Requerimientos, Divulgación de Ciencia, Tecnología e Innovación.

## **A model for the solution of non-aligned requirements: The case of ludic Software for science communication**

**Abstract:** This paper proposes methodological guidelines to be applied in the requirements engineering process when these requirements are presented as "not aligned". It is considered that this situation occurs when the software owner's objectives are different from those of its user. The software whose goal is the dissemination of science, technology and innovation, can have not aligned goals. In these cases, the end user can ignore the real purpose of the software to be used. Consequently the requirements analysis becomes crucial. The literature presents proposals to adapt the entire process of software for

educational purposes, without paying attention to the process of requirements or science communication. This proposal can be extrapolated to any Software requirements engineering process where not aligned requirements are involved.

**Keywords:** Collaborative Virtual Environments, Fictitious scenarios, Teamwork.

# 1. Introducción

La ingeniería de Requerimientos es una rama de la Ingeniería de Software que ha cobrado gran importancia debido a que representa la etapa del proceso de software en la que se minimizan los costos debido a la realización de correcciones a los productos generados por etapa. Es decir las modificaciones a los requerimientos son mucho menos costosas que las modificaciones a los programas. Un buen proceso de Ingeniería de requerimientos reduce los riesgos de inconsistencias entre las necesidades declaradas por los usuarios y las entendidas por los desarrolladores.

Bajo esta premisa podemos establecer un esquema de tres roles generales: El propietario del software, el desarrollador y el usuario. La probabilidad de éxito del software es directamente proporcional al grado en que estos tres actores coincidan en el entendimiento de los objetivos. El objetivo general de un producto de software es la premisa de la cual se derivan los objetivos particulares, es por ello que estos deben ser congruentes y consistentes.

El objetivo general de un producto de software puede ser de productividad, de entretenimiento, de aprendizaje, etc. Los usuarios generalmente están conscientes de estos objetivos, al igual que los diseñadores, programadores, propietarios, etc. A esto lo denominamos Alineación de Requerimientos.

La alineación de requerimientos debería en principio ser un requisito para iniciar el proceso de Ingeniería de requerimientos, sin embargo hay caso en

donde esta situación no se da por la misma naturaleza de la aplicación y su dominio del problema.

Este artículo reporta la investigación sobre diversos tipos de software susceptibles de presentar objetivos generales no alineados que deriven en requerimientos no alineados, y se analizan diversos tipos de software en función del dominio del problema.

Derivado de la investigación, se propone el análisis del software lúdico para la divulgación como caso de estudio, en el que se observa que de manera sistemática los objetivos del propietario del software y los del usuario son claramente diferentes.

La investigación sobre el estado del arte muestra que existen requerimientos muchas veces no explícitos pero que pueden ser cruciales para la consecución exitosa del proceso. Estos requerimientos pueden ser por ejemplo de naturaleza legal o ser tan subjetivos como los generados por las motivaciones de los propios desarrolladores. En secciones posteriores se explora la posibilidad de que estos puedan ser considerados como no alineados. Para su estudio, se analiza el caso del software para divulgación de ciencia, tecnología e innovación en sus diferentes facetas. Para ello se analiza el concepto de divulgación en la siguiente sección.

## **1.1 Divulgación**

La divulgación de Ciencia, Tecnología e Innovación es un concepto poco entendido por lo que debe ser definido con precisión. En un sentido amplio, la divulgación es un sistema de conocimiento, cuyo principio rector es la reformulación clara, amena y delimitada del conocimiento científico y tecnológico (Albourkrek, 2005), y una forma especial de transmitir dicho conocimiento (López Beltrán, 1983). De acuerdo con (Albourkrek, 2005): “La divulgación de la ciencia, la tecnología y la innovación juegan un papel crucial

en el desarrollo de la sociedad dado que erradica mitos, aumenta la calidad de vida de las personas, coadyuva a generar mejores decisiones vocacionales y propicia una mas informada participación de las personas en las decisiones colectivas”.

Los conceptos, metodologías, teorías, leyes y prácticas científicas no pueden ser comunicados con medios convencionales de manera atractiva al público en general, esto es debido a la complejidad de los mismos (Aitkin, 2005).

Esta cultura científica puede ser transmitida de manera exitosa con el uso de sistemas de cómputo con características muy particulares. Existen dos tipos de software que pueden ser útiles en la transmisión del conocimiento: De simulación y Lúdicos (Susi et al., 2007). Los sistemas de cómputo de simulación se han venido utilizando con éxito por los científicos para facilitar su propia construcción del conocimiento. Más recientemente el software lúdico con un componente de enseñanza ha sido denominado software de juegos serios. Este ha jugado un papel importante como herramienta para facilitar el aprendizaje (Gunter et al., 2006). Lo anterior es corroborado por el trabajo en (Dempsey & Johnson, 1998) quienes establecen que un juego logrará objetivos de aprendizaje si cuenta con las siguientes características: Atención, relevancia, confianza, reto, satisfacción y éxito.

Ambos tipos de software (simulación y juegos serios) han tenido éxito como coadyuvantes del aprendizaje y del autoaprendizaje.

Un sistema de cómputo que contribuya con efectividad a los objetivos de la divulgación consta de características muy particulares: Metas, propietarios, tipos de personas involucradas (stakeholders), tipos de usuarios, tipos de interacciones, proceso de desarrollo, evaluación de objetivos, etc.

Realizar acotaciones en relación al tipo de software que mejor cumpla con las expectativas de divulgación puede derivar en la maximización de sus posibilidades de éxito.

Este artículo propone una implementación sistemática de principios de Ingeniería software, (en particular de Ingeniería de requerimientos) para minimizar las dificultades que puedan presentarse por la presencia de requerimientos no alineados. Lo anterior deberá derivar en la construcción efectiva de aplicaciones de cómputo que presenten este problema.

La sección 2 muestra principios básicos de Ingeniería de requerimientos, describe la literatura relacionada con la alineación de los mismos y con la adaptación de lineamientos de dicha Ingeniería en función del tipo de aplicación a desarrollar.

La sección 3 explica el significado de la divulgación de Ciencia, Tecnología e Innovación, sus objetivos y sus motivaciones.

La sección 4 describe los lineamientos propuestos para solucionar el problema de requerimientos no alineados tomando como caso e estudio el proceso de desarrollo de software lúdico con fines de divulgación.

Finalmente las secciones 5 y 6 muestran la experiencia en el uso de esta metodología, los resultados observados, las conclusiones y el trabajo futuro propuesto.

## **2. Trabajo relacionado**

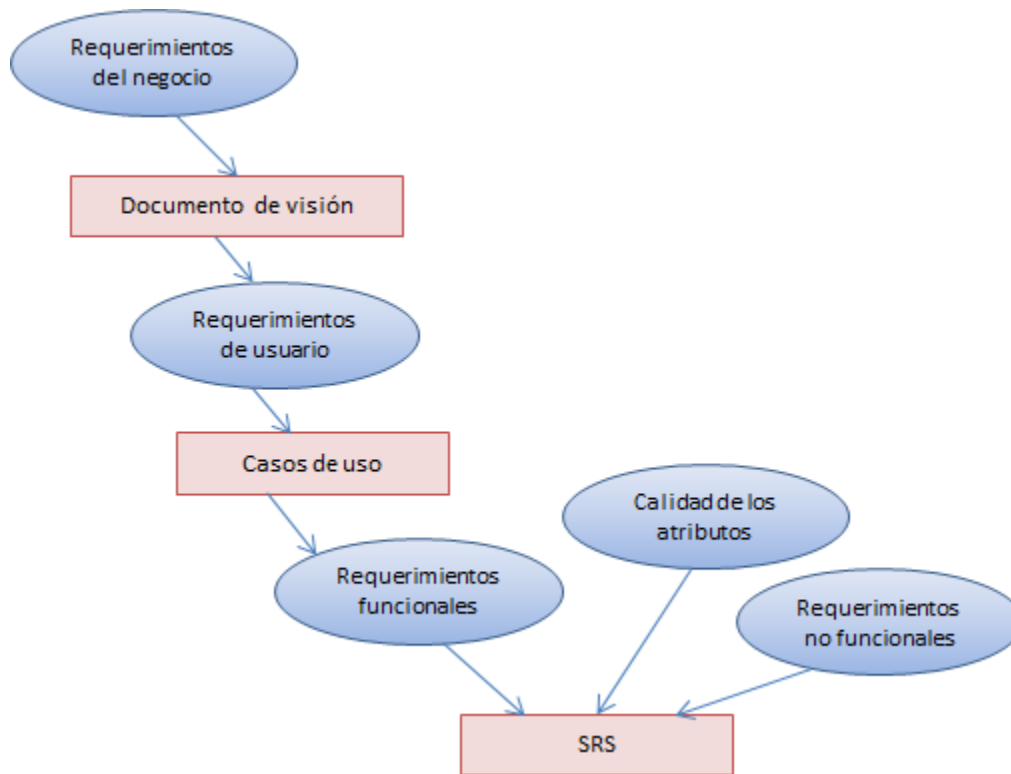
Aunque la importancia de la Ingeniería de Requerimientos es bien conocida, la literatura no muestra evidencia sobre estudios de casos en los que los requerimientos no presentan consistencia entre ellos. Esta situación se presenta cuando los stakeholders declaran objetivos con alto grado de disparidad.

Esta sección muestra los resultados encontrados en la literatura en relación al estado del arte en Ingeniería de requerimientos, las dificultades que pueden presentarse cuando los requerimientos son inconsistentes entre ellos y las

diversas adaptaciones a metodologías de requerimientos en función del dominio del problema.

## **2.1 Ingeniería de Requerimientos**

De acuerdo con la IEEE (IEEE Standard 610.12-1990) un requerimiento o requisito se define como una condición o capacidad que debe cumplir un sistema para satisfacer un contrato, un estándar, una especificación o cualquier documento formalmente prescrito. Los requerimientos deben ser: Correctos, consistentes, verificables y rastreables. La Ingeniería de Requerimientos es el proceso de licitar, entender, especificar y validar los requerimientos de los diferentes actores involucrados (stakeholders). En (Wieggers, 2000) se identifican tres niveles de requerimientos de software: Requerimientos del negocio que se reflejan en el documento de visión, Requerimientos de usuario, que se reflejan mediante un modelado (casos de uso) y Requerimientos funcionales y no funcionales documentados en las Especificaciones de Requerimientos de Software (SRS por sus siglas en inglés). Lo anterior se ilustra en la Figura 1.

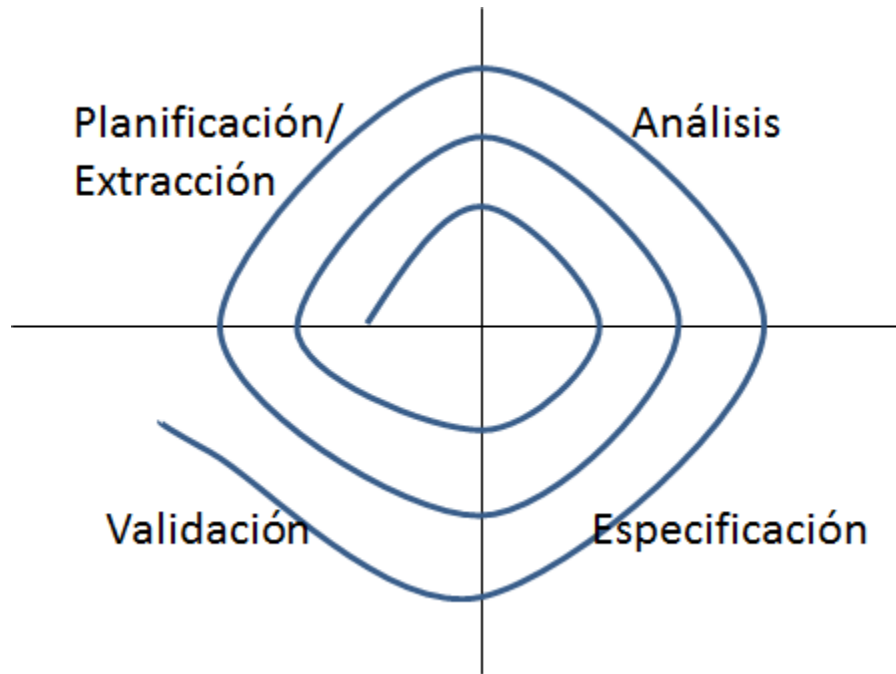


**Figura 1.**

*Proceso de Ingeniería de Requerimientos (Wiegiers, 2000)*

En (Sawyer & Kotonya, 2001) se describe el proceso de Ingeniería de Requerimientos con un modelo en espiral para cualquier tipo de software lo cual se ilustra en la figura 2. El proceso incluye las actividades de planificación/extracción, análisis, especificación y validación de requerimientos.

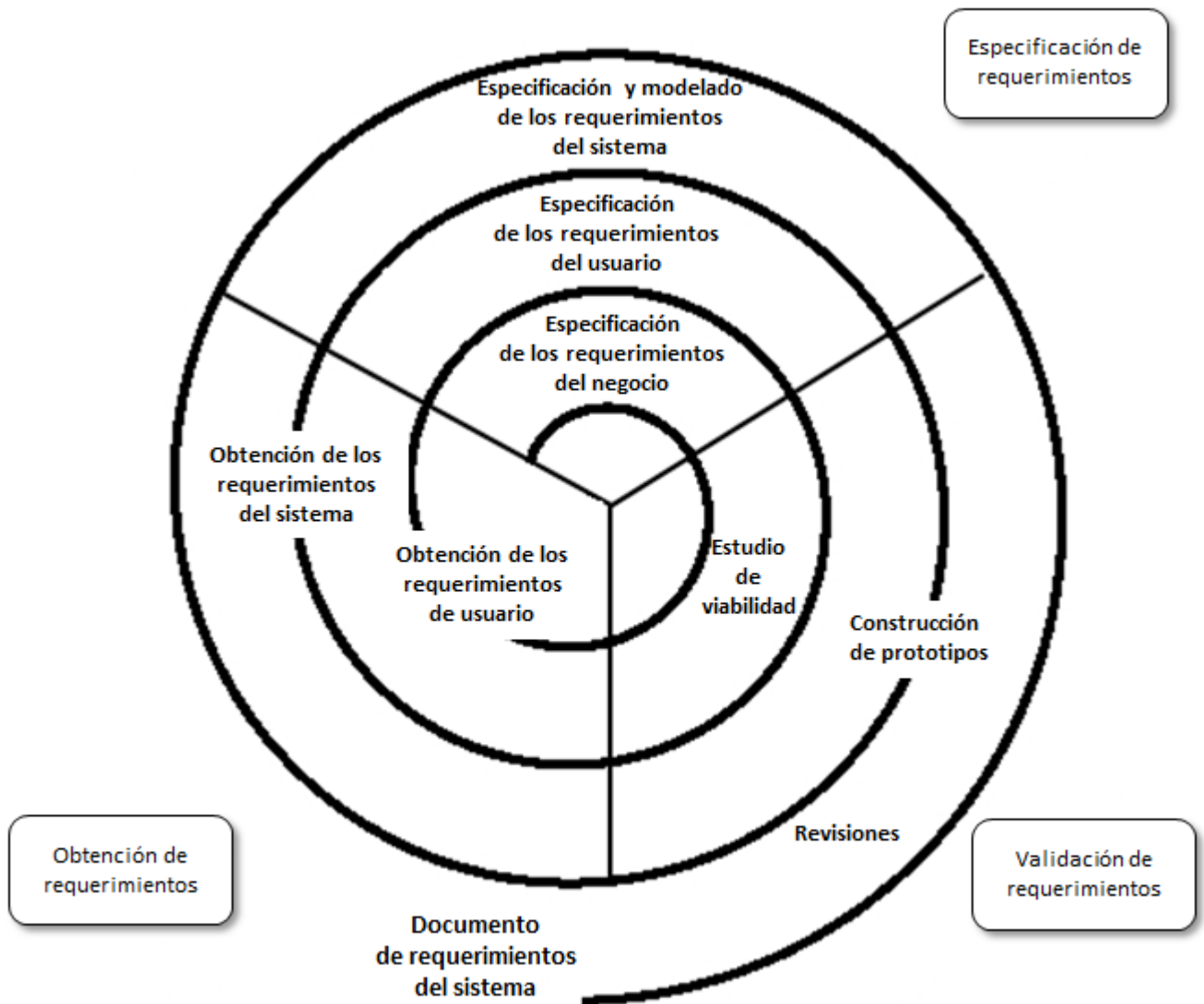




**Figura 2.**

*Proceso de Ingeniería de Requerimientos (Sawyer & Kotonya, 2001)*

De acuerdo con (Pressman, 2005) el proceso de requerimientos se compone de los siguientes pasos no necesariamente secuenciales: Concepción, obtención, elaboración, negociación, especificación, validación y gestión de requerimientos. En (Sommerville, 2010) se propone un modelo de proceso en espiral que incluye las actividades de obtención, especificación y validación de requerimientos (Figura 3).



**Figura 3.**

*Proceso de Ingeniería de Requerimientos (Sommerville, 2010)*

Como se observa, los autores coinciden en las siguientes propuestas:

1. El proceso de Ingeniería de Requerimientos debe incluir al menos las etapas de Obtención, Especificación y Validación.
2. El conjunto propuesto de participantes del proceso no presenta alta especificidad ya que del lado del cliente se propone solo los stakeholders:

Propietario e Usuario y del lado del desarrollador solo se incluye al Ingeniero de requerimientos.

3. Los entregables propuestos son al menos los documentos de Visión, Casos de Uso y Especificación de Requerimientos

Para saber si es factible adaptar el proceso de requerimientos para facilitar la construcción de software con fines específicos se presenta un resumen del estado del arte relativo al tema.

## **2.2 Alineación de Requerimientos**

Se propone la noción de Alineación de Requerimientos cuando estos son claros y consistentes en el mayor nivel de abstracción posible. Este nivel es el asociado al objetivo general del producto de software y no a los objetivos específicos de los diferentes stakeholders.

Los stakeholders de software convencional tienen objetivos pragmáticos de productividad. Por ejemplo, un software para uso de un banco (ventanillas) tiene varios objetivos: Para el que lo opera (empleado del banco), el objetivo es realizar operaciones financieras (depósitos, pagos, traspasos, etc.), esto lo hace en representación del cliente del banco (usuario indirecto). El objetivo del propietario del software (El banco) es obtener ganancias económicas sin embargo para lograrlo se sabe que será utilizado para hacer operaciones financieras. Para este tipo de aplicaciones, por ejemplo, el diseñador de las interfaces de usuario podría tener el objetivo de mostrar información de manera, precisa y estética. Como se observa, aunque las metas individuales de los diferentes roles son claramente distintas y en ocasiones hasta contradictorias, la alineación de alto nivel se presenta ya que se coincide en el objetivo general de cubrir los requerimientos de los clientes.

En el caso de un software educativo, el propietario busca que los usuarios obtengan aprendizaje a partir del uso de sus productos. El usuario, por su parte esta consiente de esa meta y busca con su uso adquirir dicho aprendizaje.

Toda la gama de stakeholders desde el usuario final hasta el productor del software, (pasando por los analistas, diseñadores, programadores, pedagogos, ilustradores, etc.) tienen una idea clara y consistente sobre el objetivo general a cumplir.

Los anteriores ejemplos muestran lo que denominamos: alineación de requerimientos. Esta se da entonces cuando los objetivos generales de los grupos involucrados muestran coincidencia.

La literatura muestra limitada información acerca de problemas de requerimientos no alineados.

En (Breux et al., 2006) se propone el concepto de Ingeniería de conformidad (Compliance Engineering). Bajo este esquema, el autor presenta una metodología que provee un enfoque sistemático para la eliminación de ambigüedades en documentos sobre leyes escritos en lenguaje natural. El artículo describe un marco de trabajo para extraer derechos y obligaciones a partir de políticas y regulaciones y un modelo para alinear estos artefactos con requerimientos de software. Este es el primer trabajo que presenta la alineación de requerimientos de software como un problema por resolver. En este caso la no alineación se da entre el conjunto de los requerimientos de usuarios y desarrolladores y entre el conjunto de los requerimientos por cumplir de leyes y regulaciones.

En (Hans, 2013) se presenta un modelo de alineación entre los requerimientos del proyecto y los requerimientos de los miembros del equipo de desarrollo del mismo. Este modelo considera las necesidades de los empleados que desarrollarán el software y las clasifica de acuerdo con el modelo jerárquico de Maslow (Taormina & Gao, 2013). Este modelo toma en cuenta necesidades de Auto-actualización, de autoestima, de seguridad y fisiológicas.

El trabajo de Hans busca coadyuvar en la búsqueda de compatibilidad en las motivaciones de las personas, sin embargo no aborda el problema que se da

cuando las motivaciones de los usuarios del software son diferentes de las que los desarrolladores tienen en relación al objetivo perseguido al usar dicho software.

No se encuentra evidencia de trabajo relacionado con el problema explícito de requerimientos no alineados de software.

De acuerdo con la investigación realizada se propone el estudio de los productos de software lúdico (software de juegos) con objetivos de divulgación o comunicación pública de la ciencia como aquel que sistemáticamente puede presentar Requerimientos No Alineados.

Las siguientes secciones exploran el estado del arte en relación a las adaptaciones propuestas al proceso de Ingeniería de requerimientos en función del dominio de los problemas que abordan.

## **2.3 Adaptaciones al Proceso de requerimientos**

La literatura muestra propuestas para adaptar el proceso de Ingeniería de Requerimientos de acuerdo al tipo de aplicaciones a desarrollar. De acuerdo con (Eljabiri & Deek) los requerimientos de proyecto pueden influenciar la decisión en relación al modelo de proceso de software a utilizar. En (Escalona & Koch, 2004) se propone la adaptación de dicho proceso al desarrollo de aplicaciones web, ellos proponen la inclusión de validación en la actividad de especificación. Su argumento en favor de proponer un enfoque diferente por el hecho de considerar aplicaciones web radica en la existencia de diferencias cruciales entre las que destacan: El gran número de stakeholders, los aspectos de navegación, la calidad del esquema (layout) de presentación y la sincronización multimedia. En (Farooq & Arshad, 2010) se propone un modelo de proceso de software específico para desarrollo de aplicaciones de web semántica, mientras que en (Shah, 2003) se centra en la etapa de diseño de este mismo tipo de aplicaciones. Estas propuestas coinciden en que la

existencia de diferencias claras en las características que rodean el desarrollo inicial del software es razón suficiente para adaptar los procesos de requerimientos para un tipo de aplicación en particular, en favor de obtener mejores resultados.

Se concluye que en caso de encontrar tipos de aplicaciones en donde sistemáticamente se presenten requerimientos contradictorios entre ellos, es válida la propuesta de un conjunto de lineamientos metodológicos que ayuden a mejorar este tipo de proyectos.

### **2.3.1 Adaptaciones al Proceso de Requerimientos para Aplicaciones de Educación**

Existen propuestas para adaptar los procesos al desarrollo de software educativo. Ejemplos de estas, incluyen la metodología de software educativo bajo un enfoque de calidad sistémica propuesta en (Díaz-Antón et al., 2008) y MeISE: una metodología de Ingeniería de Software Educativo (Figuerola, 2009). Ambas se enfocan en la producción de software para aprendizaje formal e incluyen fundamentación en teorías de educación establecidas. En relación al desarrollo de software educativo en el ámbito de la tecnología más reciente, destaca la propuesta presentada en (Cook et al., 2008), quienes proponen adaptaciones en el proceso de construcción de software móvil. En (Pérez-González et al., 2013) se propone una metodología de desarrollo de secuencias de video para la divulgación de ciencia, tecnología e innovación que incluye la programación de software para la utilización de un robot humanoide como personaje principal (COPOCYT. Robot Ruidozo), sin embargo éstas aunque son de entretenimiento, no pueden catalogarse como juegos.

## 2.3.2 Adaptaciones al Proceso de Requerimientos para Aplicaciones Lúdicas

En relación al uso de software lúdico con objetivos específicos, en (Murphy-Hill et al., 2014) se describe en qué sentidos el desarrollo de videojuegos es diferente del desarrollo de software convencional. Ellos concluyen que:

Los desarrolladores de juegos con respecto a los desarrolladores de otro tipo de software:

1. Reciben requerimientos con menor precisión.
2. Tienden a usar lo que ellos perciben como proceso ágil.

Los equipos de personas para el desarrollo de juegos le confieren mucho más valor:

1. A la creatividad
2. A las habilidades de comunicación con los no ingenieros.
3. A la confluencia de integrantes que formen un grupo más diverso.

Finalmente identifican que las personas se impresionan más con el trabajo de los desarrolladores de juegos que con el software convencional. En (Cruz-Cunha, 2012) se hace una compilación acerca de los trabajos de investigación sobre juegos serios utilizados como herramientas educativas, de negocios y de investigación. De esta compilación destaca el trabajo de (Cowley, 2014), donde se propone un modelo de proceso de software (QUARTIC) para el desarrollo de este tipo de juegos. En (Gunter et al., 2006) van más allá y proponen un paradigma formal para el diseño de este tipo de software. Finalmente en (Wilson, 2005) se destaca la existencia de un “Vibrante y variado sector del diseño, producción y distribución independiente de juegos”.

Expertos de la Industria sugieren que este tipo de desarrollo conocido como Indie gaming será cada vez más importante y creciente en el futuro inmediato. La literatura no presenta propuestas específicas de procesos para su adaptación con objetivos de divulgación. Dentro de los esfuerzos similares

destaca la propuesta en (Flanagan & Nissenbaum, 2007). Ellos proponen una metodología de diseño que considera a nivel fundamental la incorporación de valores éticos y de activismo social en los juegos de software. En (Huang & Yang, 2012) se estudian los resultados del uso de un software en línea y multi-jugador de tipo juego de rol cuyo principal objetivo es el aprendizaje incidental de vocabulario.

Finalmente en (Aitkin, 2005) se concluye que “La literatura relativa a simulaciones y juegos demuestra colectivamente que su habilidad para recrear la realidad, modelar sistemas complejos y su capacidad de visualización e interacción fomenta la práctica de la ciencia”.

La tabla 1 muestra una caracterización de las propuestas encontradas en la literatura. Como se ilustra, la mayoría de ellas propone lineamientos generales en los procesos de software, cuatro de ellas se enfocan en la etapa de diseño y solo dos se concentran en la etapa de Ingeniería de Requerimientos. Por otro lado, de las adaptaciones propuestas aunque la mitad de ellas se refieren al ámbito de aplicaciones lúdicas, solo dos consideran objetivos de divulgación. No se encontró evidencia de la existencia de propuestas centradas en la etapa de requerimientos para el desarrollo de software lúdico con fines de divulgación.

Dado que los objetivos generales de divulgación denotan la existencia de características especiales (a detallarse en la sección 3). se concluye que una propuesta como la presentada puede ser de valor.



Publicación	Proceso I-SW	Ing de. Req.	Diseño	Adaptación
Escalona y Koch [13]		✓		Web
Farooq y Arshad [14]	✓	✓		Web semántica
Shah [15]			✓	Web semántica
Díaz-Antón et al [16]	✓			Aprend. Formal
Figueroa [17]	✓			Aprend. Formal
Cook et al. [18]			✓	SW Movil
Perez-Gonzalez et al. [19]	✓			Divulgación
Cruz-Cunha [22]	✓			Juegos serios
Cowley et al. [23]	✓			Juegos serios
Gunter et al. [5]	✓			Juegos serios
Wilson [24]	✓			Video Juegos
Flanagan y Nissenbaum [25]			✓	Juegos/Activ.Social
Huang y Yang [26]	✓			Juegos/Vocabulario
Aitkin [3]			✓	Divulgación

**Tabla 1.**

*Propuestas de Procesos de software con adaptaciones*

De lo anterior se concluye lo siguiente:

1. Existen trabajos que buscan adaptar los procesos de Ingeniería de software y en particular de Ingeniería de Requerimientos a los diferentes dominios de aplicación, en particular al dominio de la educación (Díaz-Antón et al., 2008) (Figueroa, 2009) (Cook et al., 2008).
2. Existen trabajos que analizan los procesos de producción de software de juegos con fines educativos (Cruz-Cunha, 2012) (Cowley, 2014) (Wilson, 2005) (Flanagan & Nissenbaum, 2007) (Huang & Yang, 2012).
3. La literatura no presenta propuestas específicas de procesos de desarrollo de software lúdico para su adaptación con objetivos de divulgación enfocándose en la etapa de requerimientos.

### 3. Divulgación de ciencia, tecnología e innovación

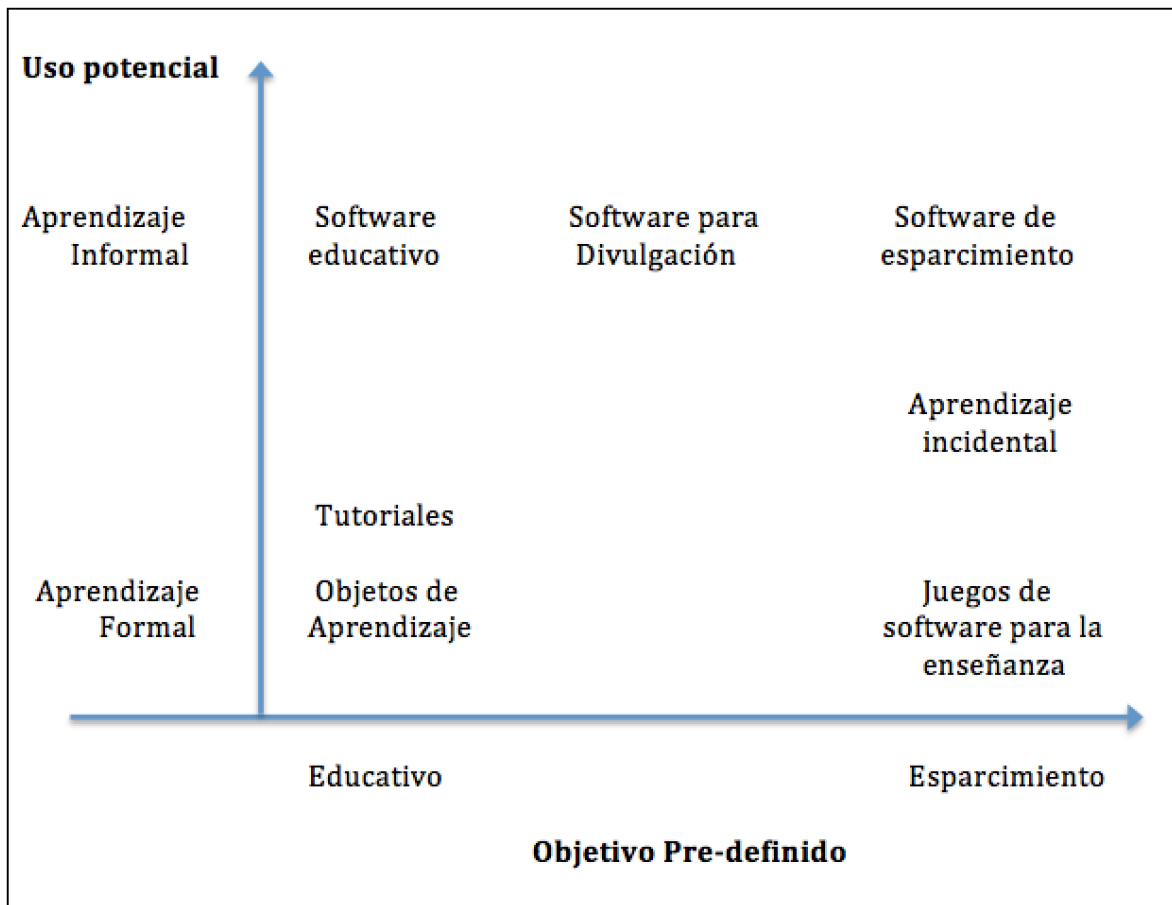
Para proponer adaptaciones al proceso de Ingeniería de Requerimientos es necesario conocer con más precisión los fundamentos y objetivos de la divulgación. Como se establece en (Sánchez Ramos & Barradas Bribiesca, 2014): “En la divulgación confluyen tres partes fundamentales: la ciencia, el divulgador y el público;” Es importante delimitar su función ya que si bien es una alternativa en la transmisión de conocimiento científico “traducido”, no pretende sustituir a la educación formal ya que no se rige bajo las mismas directrices educativas que se implementan dentro del aula, aunque esto no quiere decir que sea un proceso sin planeación, objetivos y evaluación. Ramos y Bribiesca establecen que “La divulgación no está limitada por el tiempo ni por el espacio, en ella el conocimiento fluye y es de libre elección para los usuarios”.

Para clarificar los conceptos en (Sánchez Ramos & Barradas Bribiesca, 2014) debemos también caracterizar el aprendizaje. El aprendizaje, desde un punto de vista general puede clasificarse como formal o informal. Formal es aquel que se da de manera sistemática dentro de un contexto institucional, escolarizado y bajo un esquema de evaluación. El aprendizaje informal es aquel que se da fuera del contexto referido y puede ser experimentado en cualquier otro escenario de tiempo y espacio (Cook et al., 2008). La divulgación de la ciencia, la tecnología y la innovación juega un papel crucial en el ámbito de la educación informal (Calvo Hernando, 2012).

Aunque como veremos más adelante, la divulgación no solo se circunscribe al ámbito de la enseñanza, es útil conocer los posibles diferentes objetivos que cualquier software puede tener en relación a la educación.

Se propone un esquema para caracterizar un producto de software en un espacio de dos dimensiones (figura 4). El eje horizontal representa el grado en

que los objetivos prediseñados y deliberados de un software lo colocan en una clasificación que va de totalmente educativo a totalmente de esparcimiento. El eje vertical representa el grado en que un software puede ser utilizado en procesos de aprendizaje de formal a informal.



**Figura 4.**

*Marco de referencia: Productos de software, Objetivos y usos potenciales.*

El software con el objetivo pre-definido de esparcimiento se presenta en productos que pueden contribuir a los dos tipos de aprendizaje (formal e informal); no obstante, al aprendizaje incidental se presenta en todos ellos. Los objetivos del software de divulgación son educativos y de esparcimiento pero el aprendizaje al que contribuyen es de tipo informal.

### 3.1 Objetivos de la Divulgación

A pesar del poco conocimiento que incluso la comunidad científica tiene acerca de los objetivos de la divulgación, la mayoría de los expertos y practicantes de esta disciplina coinciden en un conjunto de ellos los cuales son enumerados a continuación. La primera propuesta formal documentada acerca de la divulgación es la presentada por (Albourkrek, 2005). En ésta se establece que el fomento a la curiosidad, la imaginación y el espíritu de investigación son los objetivos a ser fomentados en el público objeto de esta actividad. Además se destaca que esta actividad puede contribuir a la orientación vocacional, a la erradicación de mitos y en general al mejoramiento de la sociedad. La tabla 2 muestra este conjunto de objetivos.

1. Es capaz de crear una atmósfera de estímulo a la curiosidad por la ciencia y su método.
2. Ayuda a despertar la imaginación.
3. Cultiva el espíritu de investigación.
4. Desarrolla la capacidad de observación, la claridad de pensamiento y la creatividad.
5. Contribuye a descubrir vocaciones científicas.
6. Propicia una relación más humana con el científico.
7. Erradica mitos.
8. Abre caminos hacia la participación del desarrollo cultural universal.
9. Enriquece la condición humana, en un sentido más filosófico.

**Tabla 2.**

*Objetivos de divulgación (Albourkrek, 2005)*

Con lo anterior se deduce que la esencia de la divulgación no está encaminada solo hacia el aprendizaje sino que tiene objetivos mucho más complejos. Estudios posteriores proponen otros grupos de objetivos. En (Shults, 2008), más recientemente, se definen los objetivos de la tabla 3.

1. Obtener recursos (fondos) para investigación
2. Entender las formas y herramientas para comunicarse con la sociedad
3. Facilitar el entendimiento público de la ciencia
4. Fomentar vocaciones científicas.

***Tabla 3.***

*Objetivos de la divulgación (Shultz, 2008)*

La tabla 3 muestra que la divulgación también contribuye a la obtención de fondos para la investigación; esto se logra debido a la premisa de que una sociedad más informada y consiente del valor de la ciencia puede influir en las decisiones que conlleven a dirigir fondos para financiarla. En este contexto podemos preguntarnos: ¿Se pueden presentar requerimientos no alineados en procesos de construcción de software lúdico para la divulgación?

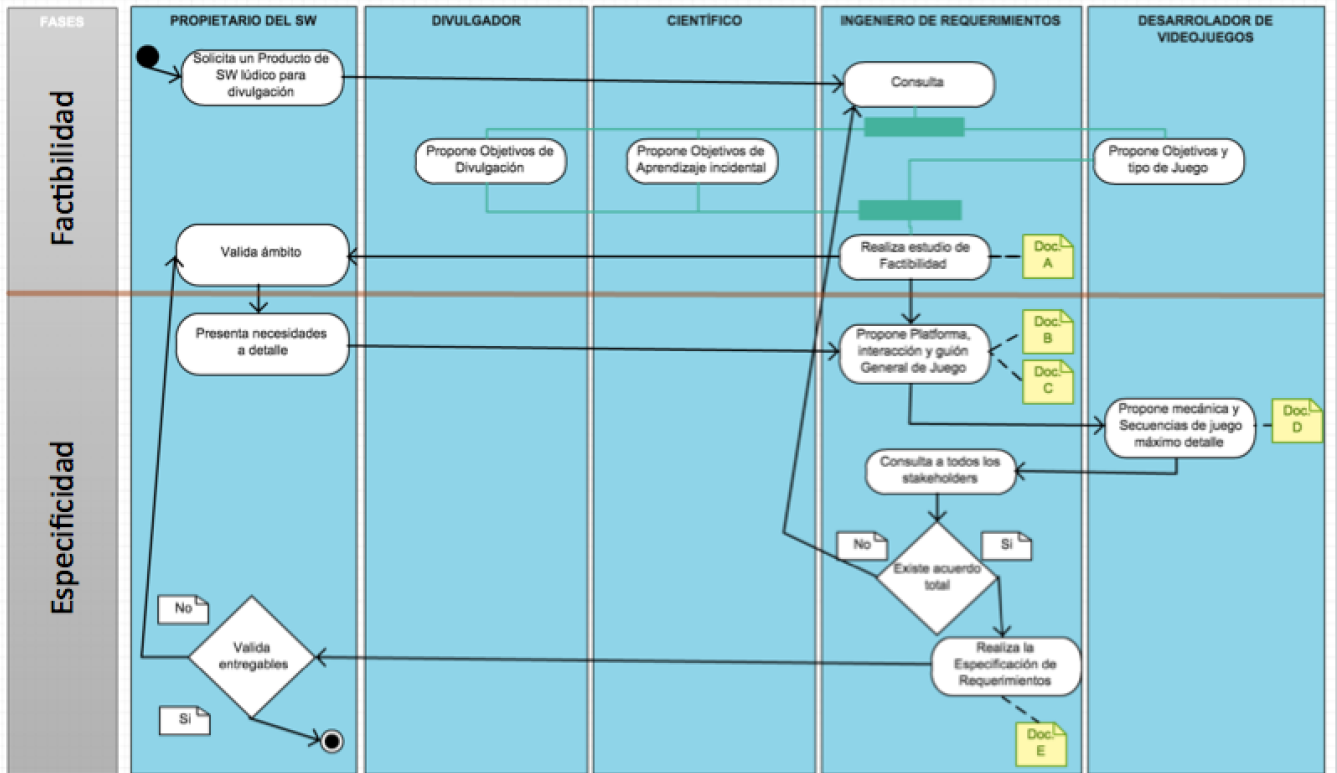
Se identifica que una característica no común pero si posible es la existencia de requerimientos con inconsistencias entre ellos.

## **4. Solución de requerimientos no alineados. Caso de estudio: Software lúdico para divulgación**

Con base en lo establecido en las secciones anteriores, podemos deducir que existen características especiales que hacen al desarrollo de software lúdico para divulgación (al cual denominaremos SWLD) diferente del desarrollo de software convencional.

Se puede afirmar que los objetivos de los stakeholders de SWLD no están alineados. El objetivo del usuario final (jugador) es simplemente la diversión, mientras que el que pagó por el desarrollo del mismo (el propietario, que es típicamente una institución gubernamental, educativa o filantrópica) tiene alguno(s) de los objetivos de las tablas 2 y 3. Por tanto y como se muestra en (Aitkin, 2005): “Los Juegos digitales son capaces de comunicar ciencia de manera encubierta, debido a que el jugador no está tratando de aprender” sino solamente de ganar el juego.

Este proceso es ilustrado en el diagrama de actividad de la figura 5.



**Figura 5.**

*Ingeniería de Requerimientos para el desarrollo de software Lúdico para la divulgación.*

En este proceso participan cinco tipos de stakeholders: propietario, divulgador, científico, Ingeniero de requerimientos y desarrollador de video juegos y se divide en dos fases: Factibilidad y Especificidad. El proceso es iniciado por el propietario del sistema, quien plantea al ingeniero de requerimientos sus necesidades iniciando así la etapa de factibilidad. Este tipo de propietarios, establecen generalmente solo una muy vaga definición del o los objetivos de divulgación perseguidos. A continuación se realiza una consulta con el divulgador, el científico y el desarrollador de video juegos (de ser posible de manera síncrona en tiempo y espacio), como resultado se genera un estudio de factibilidad (Doc. A). Este entregable debe incluir las siguientes secciones: Un documento de ámbito, uno de objetivos y uno de concepto. El documento

de ámbito debe responder las preguntas: ¿Quién está solicitando el software?, ¿quién lo usará?, ¿Qué beneficios traerá? y ¿En qué entorno se utilizará?.

El documento de objetivos es de crucial importancia para minimizar los riesgos antes referidos, éste debe incluir objetivos de divulgación y objetivos de aprendizaje incidental (ambos se constituyen como los objetivos de proyecto). Finalmente, el documento de Concepto establece el tipo de juego (Primera o tercera persona, estrategia, aventura, rompecabezas, etc.), los objetivos generales del juego y su tema general a divulgar así como su público objetivo. Estos documentos deben ser validados por el propietario del software para poder pasar a la segunda etapa. La etapa final se denomina especificidad. Esta se inicia cuando el propietario tras validar el estudio de factibilidad (Doc. A) y a la luz de éste, comunica sus necesidades a mayor detalle. Con esto el proceso garantiza la alineación de objetivos generales. El ingeniero de requerimientos prepara entonces el documento de características del sistema (Doc. B) y el Guión general del Juego (Doc. C). El Doc. B. debe precisar las plataformas en que se ejecutará el sistema (PCs, Mac, Tablet, Web, Xbox, etc.), los sistemas operativos (Linux, Windows, MacOS, IOS, Android, etc.) y muy especialmente las formas de interacción (Consola, ratón, joystick, lentes de realidad virtual, guantes, sensores, etc.). Lo anterior deberá justificarse plenamente asociándolo a los objetivos de divulgación. El Guión general del Juego (Doc. C.) debe incluir la forma de ganar, cantidad máxima y mínima de jugadores, personajes, forma de progresar y de retroceder, mecánica general de juego y duración mínima y máxima de juego. El doc. C es revisado por el desarrollador de videojuegos quien produce el Documento de Juego (Doc. D) que incluye la información de su predecesor y una muy precisa y detallada descripción del juego, con todos los posibles cursos de acción. Especialmente importante es la referencia que se debe hacer en cada especificación del juego que corresponda al o a los objetivos de divulgación que serán logrados con dicha especificación. Una vez que los cuatro documentos son producidos, estos son validados por todos los stakeholders. Con ello, el Ingeniero de requerimientos puede generar el Documento final de Especificación de Requerimientos de



Software (SRS por sus siglas en ingles) (Doc. E). El SRS es un documento común en la literatura (Sommerville, 2010). Éste incluye la funcionalidad (lo que el software debe hacer), las interfaces externas (personas, hardware y software), el desempeño (Velocidad, tiempo de respuesta, etc.), atributos (portabilidad, seguridad, mantenibilidad, etc.) y restricciones de desarrollo. Cada uno de los requerimientos tiene un indicador único y su especificidad debe ser la máxima posible, Adicionalmente se propone el uso de una matriz de rastreabilidad en donde cada requerimiento sea asociado al o a los objetivos de divulgación pre establecidos.

## **5. Experiencia de uso de metodología propuesta y resultados**

Los autores de este artículo han participado en el desarrollo de software en diversos dominios del problema, con ellos se ha identificado el tipo de aplicación que presenta invariablemente Lineamientos No Alineados, este tipo de producto es el software lúdico para divulgación.

En la primera mitad de los proyectos de divulgación desarrollados por los autores, no se siguió ninguna metodología, y por la experiencia obtenida, se llegó a la necesidad de generar una metodología basada en procesos de requerimientos, la cual ha sido aplicada a la otra mitad de los proyectos realizados por los autores.

Debido a que los propietarios de los productos generados utilizando y sin utilizar esta metodología eran los mismos, fue posible consultarlos para conocer sus opiniones sobre la utilidad de la misma. Una de los autores participó como experta en divulgación en la totalidad de los proyectos.

A cada producto de software le correspondió un tema general a divulgar (mineralogía, agua, metalurgia, tecnologías de la información, etc.).

Todos los proyectos contaron con el mismo ingeniero de requerimientos y el mismo líder de desarrollo.

Como resultado, los stakeholders manifestaron las siguientes ventajas por el hecho de usar los lineamientos propuestos:

- Mayor visibilidad del proceso de requerimientos
- Ahorro de recursos de tiempo de entre 5 a 21 %
- Reducción de la necesidad de reuniones síncronas
- Menor cantidad de iteraciones.
- Mayor rastreabilidad de los objetivos de divulgación en relación a los requerimientos específicos.

Así mismo, se enumeran las desventajas que se manifestaron:

- Inversión de demasiado tiempo en documentación
- Cuellos de botella cuando un stakeholder se demora en su trabajo.

## **6. Conclusiones y Trabajo futuro**

En este trabajo hemos presentado el estado del arte en Ingeniería de requerimientos adaptados a propósitos específicos. Así mismo se ha caracterizado la divulgación de ciencia, tecnología e innovación. Se ha observado que las características especiales del software lúdico para la divulgación, en particular la no alineación de objetivos, sugiere la necesidad de establecer un proceso particular que garantice la minimización de los riesgos de obtener malos entendidos, inconsistencias y requerimientos contradictorios. A partir de los resultados expuestos se concluye que los lineamientos aquí propuestos se constituyen como una técnica útil para la consecución de mejores resultados en la construcción de software específico para la divulgación mediante juegos.

Con esta metodología se asegura que la presencia de requerimientos no alineados no impida la consecución de un esquema para el logro de objetivos de divulgación que esté incluido a nivel fundamental. Este paradigma deberá facilitar el diagnóstico y verificación de dichos objetivos y proveer un espacio conceptual común para desarrolladores, usuarios y propietarios de productos de software.

Se propone como trabajo futuro la utilización de estos lineamientos en desarrollo de software a mayor escala (más complejo y mayor cantidad y distribución de stakeholders) y la creación de herramientas automatizadas que permitan el uso del proceso propuesto de manera más ágil y generalizada.

# Referencias

Aitkin, A. L., 2005. Playing at Reality: Exploring the potential of the digital game as a medium for science communication. Faculty of Science, The Australian National University.

Albourkrek, A., 1991. La divulgación de la ciencia. Citado en Calvo Hernando, M., 1997, Objetivos de la divulgación de la ciencia, Chasqui, 60.

Breaux, T. D., Vail, M. W., & Antón, A., 2006. Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. In Requirements Engineering, 14th IEEE International Conference, pp. 49-58. IEEE.

Calvo Hernando, M., 2012. Objetivos y funciones de la divulgación científica. ACTA, pp. 99-106.

Cook, J., Pachler, N. & Bradley, C., 2008. Bridging the gap? Mobile phones at the interface between informal and formal learning. Journal of the Research Center for Educational Technology, Vol. 4, pp. 3-18.

COPOCYT. Robot Ruidozo. Available:  
[https://www.youtube.com/results?search\\_query=%22robot+ruidoso%22](https://www.youtube.com/results?search_query=%22robot+ruidoso%22)

Cowley, B., 2014. The QUARTIC Process Model for Developing Serious Games: 'Green My Place' Case Study. Digital Da Vinci, N. Lee, Ed., ed: Springer New York, pp. 143-172.

Cruz-Cunha, M. M., 2012. Handbook of Research on Serious Games as Educational, Business and Research Tools (2 Volumes). Hershey, PA, USA: IGI Global.

Dempsey, J. V. & Johnson, R. B., 1998. The development of an ARCS Gaming Scale. *Journal of Instructional Psychology*, Vol. 24.

Díaz-Antón, M. G., Pérez M. A., Grimmán, A. C. & Mendoza, L. E., 2008. Propuesta de una Metodología de Desarrollo de Software Educativo bajo un Enfoque de Calidad Sistémica.

Eljabiri, O. & Deek, F. P. Tailoring the Software Process Model to Project requirements. *Citeseer Scientific Literature Digital Library and Search Engine*.

Escalona, M. J. & Koch, N., 2004. Requirements engineering for web applications-a comparative study. *J. Web Eng.*, Vol. 2, pp. 193-212.

Farooq, A. & Arshad, M. J., 2010. A Process model for Developing Semantic Web Systems. *New York Science Journal*, Vol. 3, pp. 43-39.

Figueroa, M. A. A., 2009. MeISE: Metodología de Ingeniería de Software Educativo. *Revista Internacional de Educación en Ingeniería*, Vol. 2.

Flanagan, M. & Nissenbaum, H, 2007. A game design methodology to incorporate social activist themes. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 181-190. ACM.

Gunter, G. A., Kenny, R. F. & Vick, E. H., 2006. A Case for a Formal Design Paradigm for Serious Games. In *Human Systems, Digital Bodies*.

Hans, R., 2013. A Model for Aligning Software Projects Requirements with Project Team Members Requirements.

Huang, B. G. & Yang, J. C., 2012. A Multiplayer Online Role-Playing Game for Incidental Vocabulary Learning. In Proceedings of the 20th International Conference on Computers in Education (ICCE 2012), Singapore.

IEEE Standard Glossary of Software Engineering Terminology, 1990. IEEE Std 610.12-1990, pp. 1-84.

López Beltrán, C., 1983. La creatividad en la divulgación de la ciencia. *Naturaleza*, Vol. 14.

Murphy-Hill, E., Zimmermann, T. & Nagappan, N., 2014. Cowboys, ankle sprains, and keepers of quality: how is video game development different from software development?. In Proceedings of the 36th International Conference on Software Engineering, pp. 1-11. ACM.

Pérez-González, H. G., García, R. M. M. & Cuervo, F. D. C., 2013. Metodología para el desarrollo de proyectos de divulgación utilizando un robot humanoide. XIX Congreso Nacional de Divulgación de la Ciencia y la Técnica, Zacatecas, México, pp. 649-658.

Pressman, R. S., 2005. *Software Engineering: a practitioner's approach*. McGraw-Hill International Edition.

Sánchez Ramos, M. E. & Barradas Bribiesca I., 2014. Divulgación de la Ciencia a través de los dispositivos móviles. *Alternativa Educativa en México*. Congreso Virtual sobre Tecnología, Educación y Sociedad.

Sawyer, P. & Kotonya, G., 2001. Software Requirements. In IEEE SWEBok Project Report. 34, ed: Schwabe, D., Rossi, G.

Shah, A., 2003. OODM: an object-oriented design methodology for development of web applications. Information modeling for internet applications, ed: Patrick van Bommel, pp. 189-229.

Shults, A., 2008. Objectives and tools of science communication in the context of globalization. Doktors der Philosophie, Universität des Saarlandes, Saarbrücken.

Sommerville, I., 2010. Software Engineering. Addison-Wesley, 9th edition.

Susi, T., Johannesson, M. & Backlund, P., 2007. Serious Games – An Overview. University of Skövde HS-IKI-TR-07-001

Taormina, R. J. & Gao, J. H., 2013. Maslow and the motivation hierarchy: Measuring satisfaction of the needs. The American journal of psychology, 126(2), pp. 155-177.

Wieggers, K. E., 2000. Karl Wieggers Describes 10 Requirements Traps to Avoid. Software Testing and Quality Engineering, Vol. 2

Wilson, J., 2005. Indie Rocks! Mapping Independent Video Game Design. Media International Australia, Incorporating Culture & Policy, Vol. 5, pp. 109-122.

## Notas biográficas:



**Héctor G. Pérez González** Obtuvo el grado de Maestro en Ciencias Computacionales en la Universidad Nacional Autónoma de México in 1993 y el de Doctor en Ciencias Computacionales en la Universidad de Colorado en 2003. Sus áreas de interés son la Ingeniería de Software y la Interacción Humano Computadora.



**Rosa M. Martínez-García** Licenciada en Ciencias de la Comunicación por la Universidad del Centro de México (UCEM). Se desempeñó como Coordinadora de las Licenciaturas en Comunicación Gráfica y Ciencias de la Comunicación de la UCEM de 1996 a 1999 y como Directora de Divulgación del Consejo Potosino de Ciencia y Tecnología (COPOCYT) de 2010 a 2013. Es Miembro Fundador de la Red de Divulgación de Ciencia Tecnología e Innovación (REDICITI) del estado de San Luis Potosí. Actualmente, es Jefa del Departamento de Difusión y Divulgación “D3”, de la Facultad de Ingeniería de la UASLP. Responsable del Programa de Divulgación Ingenialidades y coordinadora del grupo “ingeniosos divulgando”.



**Francisco E. Martínez-Pérez** Obtuvo el Título de Ingeniero en Computación en la Facultad de Ingeniería de la Universidad Autónoma de San Luis Potosí (UASLP) en 2001, el grado de Maestro en Ciencias Computacionales por la UASLP in 2005 y el grado de Doctor en Ciencias en la Universidad Autónoma de Baja California en 2012. Actualmente es administrador del laboratorio UDICEI. Sus áreas de



interés son la Interacción Humano Computadora, Ingeniería de Software, Cómputo ubicuo y procesamiento de imágenes.



**Sandra E. Nava-Muñoz** Obtuvo el Título de Ingeniero en Computación en la Facultad de Ingeniería de la Universidad Autónoma de San Luis Potosí (UASLP) en 2001, el grado de Maestro en Ciencias Computacionales por la UASLP in 2005 y el grado de Doctor en Ciencias en la Universidad Autónoma de Baja California en 2013. Sus áreas de interés son la Interacción Humano Computadora, Ingeniería de Software y Cómputo ubicuo.



**Alberto Nuñez-Varela** Obtuvo el Título de Ingeniero en Computación en la Universidad Autónoma de San Luis Potosí (UASLP) en 2005, el grado de Maestro en Ingeniería en Computación en el Centro de Investigación y Estudios de Posgrado en la UASLP en 2011. Actualmente es estudiante del programa de Doctorado en Ciencias Computacionales y profesor en la Universidad Autónoma de San Luis Potosí. Su área de interés es la Ingeniería de Software, en particular, Calidad de Software y Métricas de software.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.