
Mecanismo para la generación sistemática de pruebas funcionales de smart contracts en sistemas de gestión de publicaciones digitales

Mechanism for the systematic generation of smart contract functional tests of in digital publication management systems

**Nicolás SÁNCHEZ-GÓMEZ (1), Javier J. GUTIÉRREZ (1),
Enrique E. PARRILLA (2), Julián A. GARCÍA-GARCÍA (1)**

(1) Departamento de Lenguajes y Sistemas Informáticos, ETSII, Universidad de Sevilla, Avenida Reina Mercedes s/n, 41012, Sevilla, {nicolassg, javier, juliangg}@us.es.

(2) Lantia Publishing S.L., Plaza de la Magdalena, num. 9, Planta 3ª, 41001 Sevilla, enrique@lantia.com

Resumen

La tecnología *blockchain* ha ganado un protagonismo significativo en el mundo de los negocios. Su impacto se ha sentido en numerosos sectores, pero su integración e interoperabilidad continúa siendo compleja. Siguen existiendo numerosos retos, tanto para el área usuaria y consultores de negocio, como para los ingenieros informáticos. Uno de los retos identificados por la comunidad investigadora, es la necesidad de proporcionar mecanismos que permitan especificar, verificar y validar los requisitos y reglas de negocio que deben cumplir los *smart contracts* antes de su despliegue en una red *blockchain*. Este artículo describe una propuesta basada en la ingeniería dirigida por modelos y orientada al usuario, cuyo objetivo es obtener pruebas funcionales a partir de las especificaciones de los *smart contracts*, de forma sistemática. Esta propuesta ha sido validada en el proyecto SmartISBN, proyecto I+D+i, que tenía entre sus objetivos asegurar la calidad software de los *smart contracts* y, sobre todo, mejorar la trazabilidad de las publicaciones digitales (libros y revistas electrónicas). utilizando para ello la tecnología *blockchain*. En este contexto, la presente propuesta, ha facilitado la comunicación entre expertos funcionales (autores, editoriales, libreros, etc.) e ingenieros informáticos durante la etapa de especificaciones de la solución global SmartISBN y, de los *smart contracts*, en particular, así como durante la definición de las pruebas funcionales necesarias para la validación del proyecto.

Palabras clave: Publicaciones digitales. Pruebas de software. Blockchain. Smart contracts. Trazabilidad. ISBN.

1. Introducción

La producción editorial abarca todas aquellas actividades destinadas a la transformación de una obra creativa en una publicación impresa o digital hasta que llega a su público objetivo. Esta producción viene precedida por la adquisición de los derechos de autor de la obra creativa.

Abstract

Blockchain technology has gained significant prominence in the business world. Its impact has been felt in many sectors, but its integration and interoperability remain complex. Many challenges remain, both for users and business consultants, as well as for IT engineers. One of the challenges identified by the research community is the need to provide mechanisms to specify, verify and validate the requirements and business rules that smart contracts must comply with before they can be deployed in a blockchain network. This paper describes a proposal, based on model-driven and user-oriented engineering, that aims to obtain functional verification from smart contract specifications in a systematic way. This proposal has been validated in the SmartISBN project, an R+D+i project whose objectives included ensuring the software quality of smart contracts and, above all, improving the traceability of digital publishing (electronic books and journals) using blockchain technology. In this context, this proposal has facilitated the communication between functional experts (authors, publishers, booksellers, etc.) and IT engineers during the specification phase of the global SmartISBN solution and in particular of the smart contracts, as well as during the definition of the functional tests necessary for the validation of the project..

Keywords: Digital publishing. Software testing. Blockchain. Smart contracts. Traceability. ISBN.

Otras partes del proceso editorial son la edición, el diseño gráfico, la impresión o publicación digital, etc., siendo necesario durante todo este proceso hacer un uso intensivo de metadatos y datos de la obra literaria. Esto hace que la calidad de las actuaciones llevadas a cabo esté muy ligada a la cantidad y calidad de estos metadatos y datos.

El proceso de producción, junto con el de distribución de publicaciones digitales (libros y revistas electrónicas, entre otros) es un proceso complejo (Martínez Alés, 2001). Además, de este proceso forman parte una gran variedad de actores, cada uno de ellos con necesidades distintas.

Una obra, una vez escrita, entra en un proceso de producción que puede tardar semanas o incluso meses y requiere una inversión económica importante. Si bien, existen mecanismos de publicación bajo demanda con plazos de entrega de días (que no ofrecen una calidad asimilable), normalmente, quedan relegados a nichos de mercado muy específicos. Después de la producción, los distribuidores llevan las publicaciones desde las editoriales hasta los puntos de venta o de distribución. Estos puntos pueden ser librerías físicas, plataformas de internet o ambas (Magañán-Díaz et al., 2020).

La aparición de nuevas tecnologías que permiten el almacenamiento masivo de información o nuevas tecnologías que permiten el registro de información de forma segura y transparente, sin intermediarios, como es la tecnología *Blockchain* (Alharby et al., 2018) (Chirinos et al., 2019) (Gramoli, 2022), abre la posibilidad de ofrecer alternativas novedosas dentro de la industria editorial. En este sentido, la tecnología *blockchain* y, sobre todo, *los smart contracts* (Zou et al., 2019) pueden realizar aportaciones de gran valor añadido en el ámbito de las publicaciones digitales. Por ejemplo, algunos autores (Safdar et al., 2022) analizan los beneficios que la tecnología *blockchain* puede llegar a ofrecer, como es tener más transparencia, integridad, seguridad y eficiencia en el seguimiento de las obras en cada etapa del proceso de producción de éstas.

Asimismo, existen multitud de trabajos relacionados (Zeng et al., 2019; Yuan et al., 2022) en los cuales se utilizan la *blockchain* y los *smart contracts* para el seguimiento del proceso de cadena de suministro (*supply chain*). Sin embargo, el uso de *smart contracts* presenta aún retos en cuanto a la especificación de requisitos (Sánchez-Gómez et al., 2020) que deben cumplir estos “contratos digitales”, así como en cuanto a la generación de pruebas funcionales que permitan verificar y validar el buen funcionamiento del *smart contract* a partir de su propia especificación (Sánchez-Gómez et al., 2020).

El objetivo del presente artículo es proponer una solución basada en la ingeniería dirigida por modelos (Bézivin, 2004) que ofrezca mecanismos para especificar *smart contracts* y que, a partir de estas especificaciones, sea posible generar de manera sistemática las pruebas funcionales de estos, al objeto de verificar y validar que el

software funciona de la forma esperada en todos los pasos del proceso.

Esta propuesta, además, ha sido validada dentro del proyecto SmartISBN (The Medizine, 2021), cuyo propósito era obtener una plataforma web sobre *blockchain* que permita gestionar el proceso de producción y distribución de obras literarias facilitando que todas las partes involucradas (editoriales, distribuidores, autores y agencias) puedan tener información instantánea sobre el estado y situación comercial de cualquier lanzamiento.

En este contexto, la propuesta teórica que se presenta ha sido utilizada para especificar los requisitos funcionales del sistema completo y, en particular, los acuerdos, condiciones y reglas que deben cumplir los *smart contracts*, así como para generar las pruebas funcionales de estos *smart contracts*.

La organización de este trabajo se describe a continuación. La sección 2 presenta el contexto del proyecto SmartISBN. La propuesta de solución junto con el caso de validación se describe en la sección 3. Finalmente, las conclusiones y trabajo futuro se presentan en la sección 4.

2. Contexto: Proyecto SmartISBN

El proyecto SmartISBN (1) fue abordado por la empresa Lantia Publishing, en colaboración con el grupo de investigación ES3 de la ETSII (Universidad de Sevilla). El proyecto tenía como objetivo obtener un prototipo, no comercializable, de una plataforma basada en la tecnología *blockchain*, que permitiese gestionar los metadatos y datos de las obras literarias y hacer un seguimiento de la trazabilidad de estas obras durante el proceso de producción y distribución.

Para la metadatación de obras literarias ya existen estándares para la estructuración de sus metadatos, pero la realidad es que cada editorial sigue y sigue utilizando su propia estructura de datos, es decir, cada editorial utiliza su propio sistema de gestión con un conjunto mínimo básico de datos a su medida, lo cual disminuye la potencia de gestión que se podría aplicar.

En este contexto, el proyecto SmartISBN abordó el problema indicado, mediante la investigación y desarrollo de un modelo de metadatos aplicable al sector editorial, es decir, que tuviera en cuenta el estándar ONIX (2), y que permitiera el tratamiento de los datos y metadatos asociados a una publicación de forma unificada, de tal manera que fuera posible gestionar el estado de cada obra, de forma individual, durante las etapas del proceso de producción y distribución. En

concreto, este proyecto pretendía alcanzar los siguientes objetivos:

- Obtener y almacenar, en un repositorio único, la información de obras literarias, de manera automática o semiautomática. Esta automatización consistía en obtener datos, de obras literarias publicadas en internet, mediante herramientas o utilidades que rastrean y capturan información sobre las mismas en sitios web.
- Realizar el tratamiento de la información obtenida, teniendo en cuenta las necesidades específicas de cada uno de los actores involucrados: editoriales, distribuidoras, librerías, etc. Además, esta información debía ser almacenada en metadatos, siguiendo el estándar ONIX y debía ser autoverificable, es decir, en caso de incongruencias el sistema debería reportar las posibles incidencias.
- Facilitar el seguimiento de las distintas actividades que se realicen con las obras literarias mediante el registro de estas actuaciones en una red *blockchain*. Este objetivo permite que haya evidencias de todas estas actividades sin posibilidad de alteraciones, gracias a las características intrínsecas de la *blockchain*. Esto permitía asegurar la integridad, transparencia, seguridad e inmutabilidad de las transacciones realizadas y facilitar la detección de posibles

fraudes u actuaciones anómalas de una manera más sencilla y eficiente.

3. Propuesta de solución

Esta sección describe nuestra propuesta para obtener modelos de *smart contract*, a partir de las especificaciones indicadas por área usuaria o cliente y generar, de forma sistemática, modelos de pruebas funcionales a partir de dichos modelos de *smart contracts*. Concretamente, la Sección 3.1 describe cuáles son los componentes teóricos de nuestra propuesta y, la Sección 3.2, describe cómo ha sido aplicada y validada dicha propuesta en el marco del proyecto SmartISBN.

3.1. Propuesta de generación de pruebas funcionales a partir de los modelos de *smart contracts*

La propuesta que se describe a continuación (Figura 1), está basada en los principios de las pruebas tempranas (Jain, 2016), cuyo objetivo es generar pruebas funcionales lo antes posible, en concreto, a partir de las especificaciones funcionales de la solución global y, en nuestro caso, las especificaciones, reglas y condiciones que deben cumplir los *smart contract* a implementar — todo ello de manera independiente a la plataforma *blockchain* a emplear—.

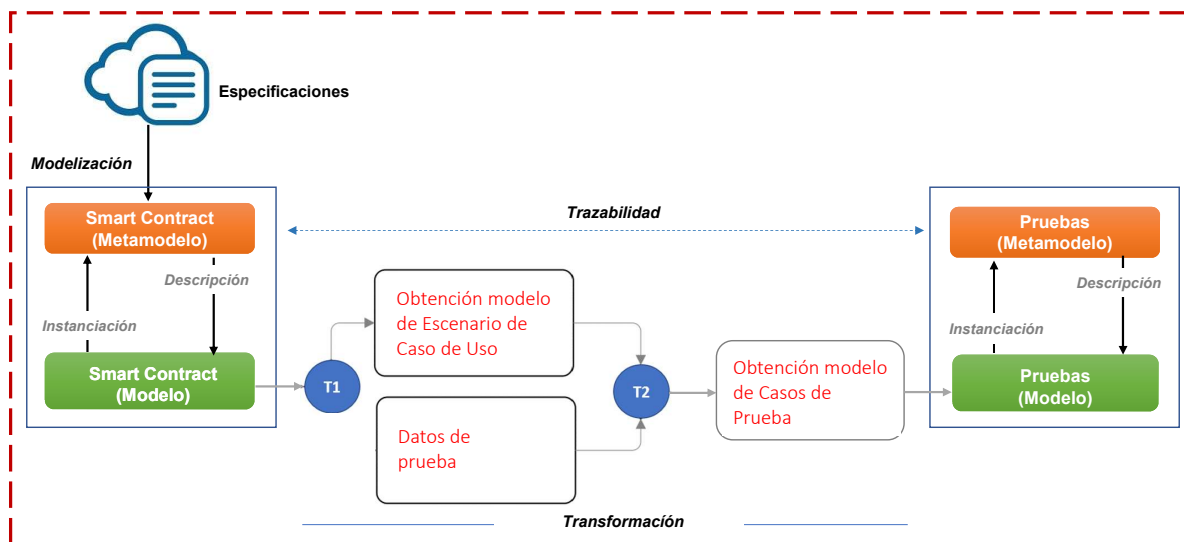


Figura 1. Propuesta de generación de pruebas funcionales a partir de los modelos de smart contracts

Para este propósito, nuestra propuesta (ver Figura 1) pone el foco en el paradigma de la ingeniería dirigida por modelos (Bézivin, 2004), definiéndose para ello:

- Un metamodelo de *smart contract* (Figura 2) que contiene la definición de los conceptos y
- Un metamodelo de pruebas funcionales (Figura 3), que contiene la definición de los

relaciones necesarias para obtener un modelo de *smart contracts* a partir de especificaciones funcionales, acuerdos, reglas y condiciones indicadas por el área usuaria o cliente.

conceptos y relaciones necesarias para obtener un modelo de pruebas funcionales.

- Unos mecanismos —las transformaciones T1 y T2 indicadas en la Figura 1— que permitan

generar un modelo de pruebas funcionales a partir del modelo de smart contracts, teniendo en cuenta las reglas de transformación entre metamodelos.

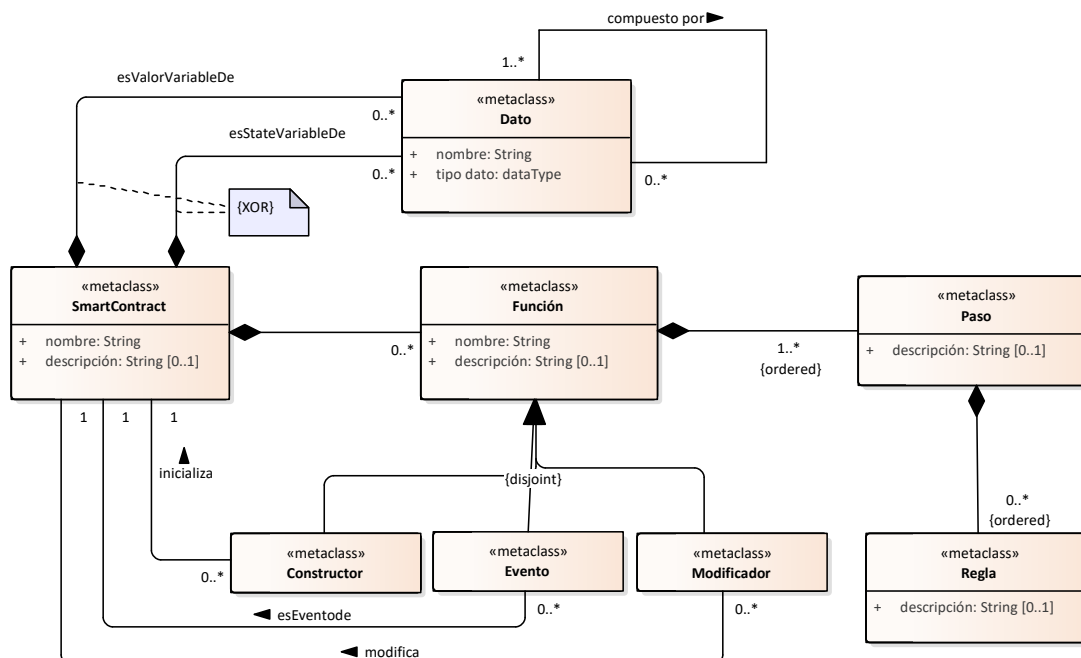


Figura 2. Propuesta de metamodelo de smart contract

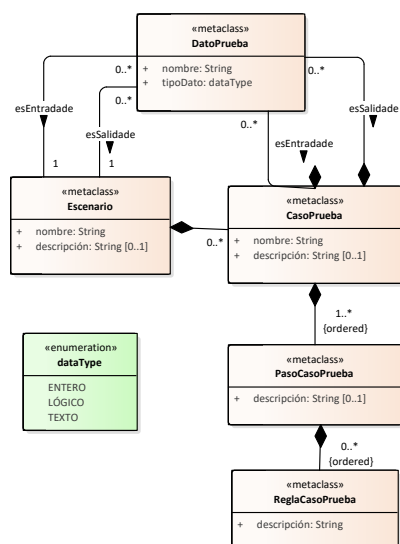


Figura 3. Propuesta de metamodelo de pruebas funcionales

Nuestra propuesta de metamodelo de *smart contract* (3), tiene como concepto central la metaclass «SmartContract». Alrededor de ésta, se describen el resto de elementos y relaciones de

un *smart contract*. En concreto, el *smart contract* está compuesto por un conjunto de datos de negocio (metaclass «Dato») que representa cualquier información relativa a los actores que intervienen durante la ejecución del contrato, así como los datos (tanto internos como externos) que se pueden tratar y gestionar. El tratamiento de estos datos es realizado por la metaclass «Función», la cual representa cualquier conjunto de acciones que pueden ser realizadas por el *smart contract* sobre los datos en base a un conjunto de pasos (modeladas con la metaclass «Paso») y a posibles restricciones (modeladas con la metaclass «Regla»).

Es importante mencionar que las restricciones modelan los términos, reglas y condiciones del *smart contract*, imponiendo los criterios sobre cuándo se puede realizar una acción, es decir, si las circunstancias permiten o no ejecutar la acción. Por tanto, en un modelo de *smart contract*, una restricción vincula la ejecución de una determinada acción al cumplimiento de dichas condiciones y reglas.

Siguiendo el esquema de la Figura 1, se describe brevemente a continuación el mecanismo para generar pruebas funcionales de forma sistemática a

partir de los modelos de *smart contract* especificados. En concreto, partiendo de los modelos del *smart contract* obtenidos conforme al metamodelo de *smart contract* (Figura 1), se establecen las siguientes transformaciones para generar escenarios de caso de uso y, por cada escenario, los casos de prueba:

- La primera transformación (T1) permite obtener un modelo de escenario a partir de un modelo de *smart contract*. Es decir, se genera un modelo de escenario conforme al metamodelo de pruebas funcionales, partiendo del modelo de *smart contract* —modelo que se obtuvo conforme al metamodelo de *smart contract*—.
- La segunda transformación (T2) permite obtener un modelo de caso de prueba a partir de un modelo de escenario, conforme al metamodelo de pruebas funcionales y, teniendo en cuenta los posibles datos de prueba.

3.2. Aplicación y validación de la propuesta

En esta sección, por un lado, se expone uno de los procesos o flujos de trabajo que se ha tenido en cuenta en el marco del proyecto SmartISBN y se describe, de forma ilustrativa, la arquitectura tecnológica y funcional implementada. Y, por otro lado, se presenta como se ha aplicado nuestra propuesta teórica, descrita en la Sección anterior, al objeto de generar, de forma sistemática, las pruebas funcionales a partir de que la especificación del sistema global y de las especificaciones particulares de los *smart contracts*.

3.2.1. Proceso de producción y distribución de una obra literaria

La Figura 4 representa algunas etapas del proceso de producción y distribución de una publicación, así como los diferentes actores que intervendrían en cada etapa. Para ello, se emplea la notación de diagrama de secuencia UML (*Unified Modeling Language*) (Fontela, 2012) que nos permite representar el flujo de trabajo entre actores.

Inicialmente, se podría considerar que el proceso de distribución comienza con una primera etapa «E1. Concepción y obtención de la publicación», en la que su Autor le da forma, consistencia y sentido a su obra hasta que obtiene el manuscrito final. A continuación, el Autor iniciaría la segunda etapa: «E2. Procesamiento editorial de la publicación». En esta etapa, el Editor recibe el manuscrito y lleva a cabo su proceso de revisión, catalogación de la publicación dentro de su línea editorial e identificación de metadatos. Una vez completado este procesamiento, el Editor iniciaría la etapa «E3. Impresión y distribución», estableciendo diferentes contratos u órdenes de

pedido con la empresa Distribuidora para que ésta comience la impresión física y/o difusión digital de las diferentes ediciones de la publicación. Finalmente, el proceso finalizaría con la etapa «E4. Adquisición de ejemplares de la publicación», en la que Librerías (u otros puntos de venta) deben verificar la entrega de las publicaciones bajo distribución con el pedido realizado.

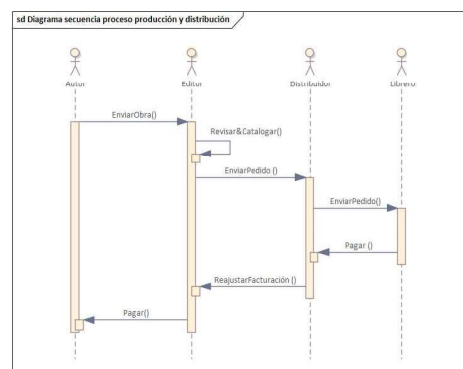


Figura 4. Diagrama de secuencia asociado al proceso de producción y distribución

Considerando el proceso anterior, cabe resaltar que durante las transiciones entre las diferentes etapas se llevan a cabo pedidos, pagos, entregas, etc., entre los diferentes actores que intervienen en el proceso. En este sentido, en el contexto del proyecto SmartISBN, resulta crucial mantener la trazabilidad de todas estas transacciones durante el proceso completo de la cadena de producción y distribución.

3.2.2. Arquitectura de la plataforma SmartISBN

La arquitectura tecnológica y los subsistemas que componen la plataforma SmartISBN se muestran en la Figura 5.

La plataforma está compuesta por una capa de Administración para que, los usuarios con rol “administrador”, puedan llevar a cabo la gestión de usuarios, roles y permisos de acceso a la plataforma, así como el control del estado y los componentes de la plataforma. Incluye, además, una capa de *Interoperabilidad* que facilita la comunicación entre los diferentes subsistemas y la *blockchain*.

A nivel de subsistemas, la plataforma cuenta con el subsistema de Catalogación. Éste es el encargado de automatizar el proceso de ingesta y análisis de metadatos. Otros subsistemas son el sitio web (front-end) y el terminal punto de venta, en el que se gestiona, respectivamente, el repositorio de publicaciones digitales y la gestión del inventario y ventas.

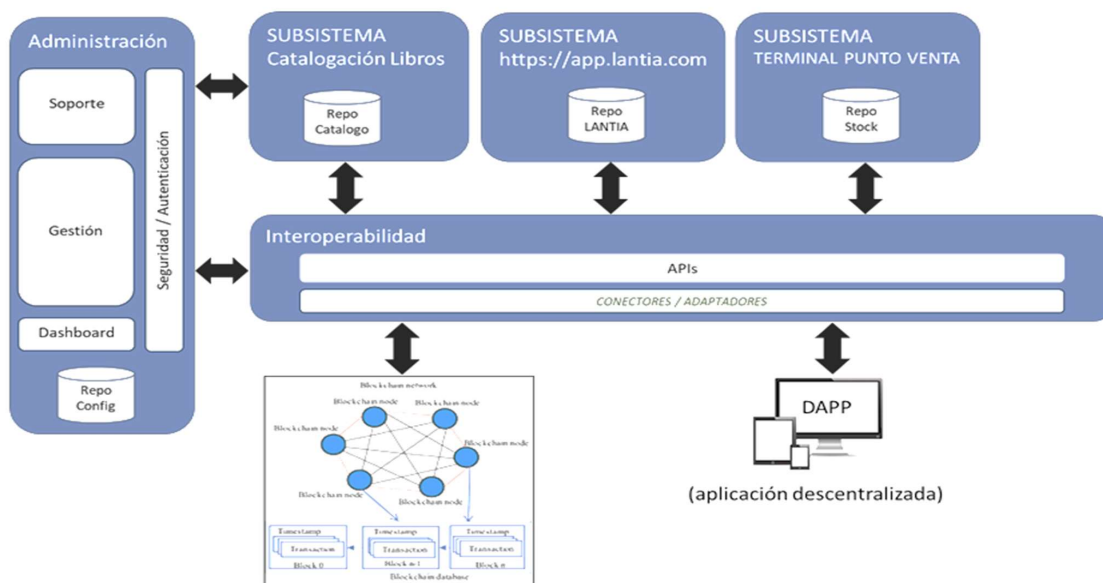


Figura 5. Arquitectura de la plataforma SmartISBN

3.2.3. Aplicando la propuesta de generación sistemática de pruebas funcionales de smart contracts

En el contexto del proyecto SmartISBN, fue necesario implementar varios smart contracts teniendo en cuenta los requisitos funcionales, tanto a nivel de registro, validación o consulta (Figura 6).

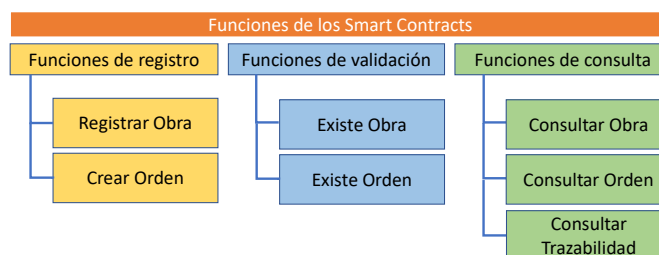


Figura 6. Smart Contract (funciones)

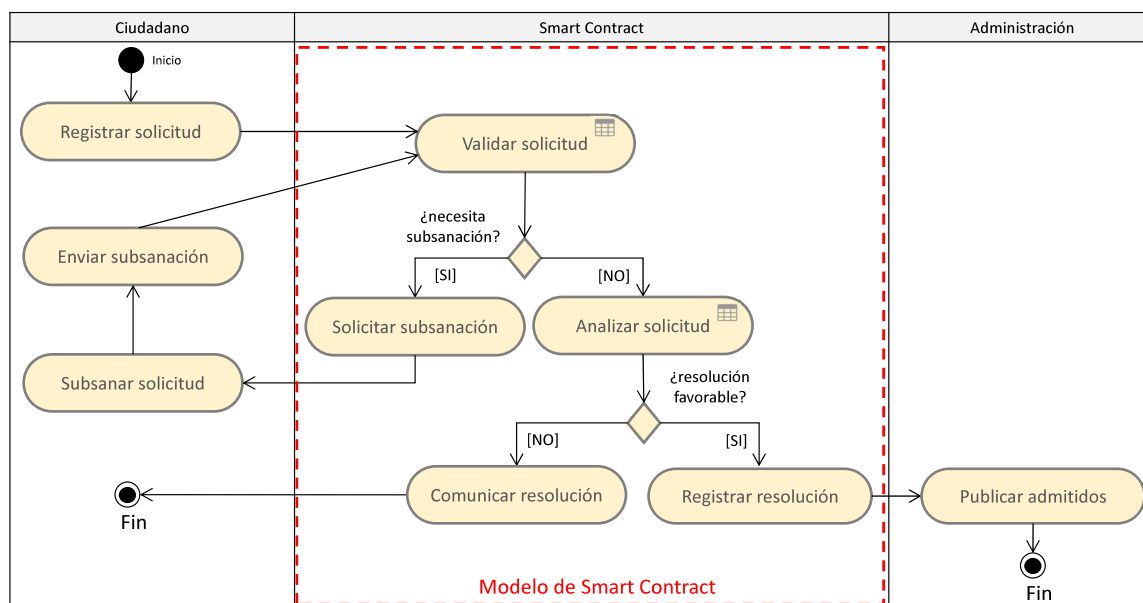


Figura 7. Diagrama UML de actividad

En el presente artículo nos centraremos en un *smart contract*, dadas las limitaciones de espacio de este artículo. Y, por el mismo motivo, la casuística completa de generación de pruebas funcionales tampoco podrá ser descrita minuciosamente y, a modo de ejemplo, solo se usará el siguiente diagrama UML de actividad (Figura 7) que representa el caso de uso “Registrar una Obra”.

Este diagrama de actividad no es más que un flujo de trabajo donde se describe el comportamiento de un sistema o proceso de negocio. En nuestro caso, este diagrama ha estado guiado por el metamodelo de *smart contract* descrito en el apartado anterior y, como se puede apreciar en la actividad “Registrar petición”, aparece un pequeño icono en forma de tabla. Éste indica que hay reglas y restricciones a tener en cuenta antes de avanzar a la siguiente actividad.

Para modelar las reglas y restricciones, se ha utilizado el estándar DMN, del inglés *Decision Model Notation* (4), que se apoya en tablas de decisión donde cada fila es una regla (ver Figura 8).

Tabla de decisión			
U	Idioma	Nº páginas	SALIDA
	Inglés, Español	Nº Entero	SI / NO
1	SI	< 50	SI
2	SI	[50, 100]	NO
3	NO	[100, 150]	SI
4	SI	> 150	NO

Figura 8. Tabla de decisión (Diagrama DMN)

Si observamos la Figura 8, las reglas y restricciones se apoyan en atributos y sus posibles valores. Por ejemplo, si el idioma de la obra es inglés o español y tiene menos de 50 páginas, ésta puede ser recepcionada para ser validada por el editor.

Con estos diagramas ya tendríamos modelada la funcionalidad del sistema y, en particular las reglas y restricciones del *smart contract*.

En el contexto de pruebas funcionales, un escenario es una ruta específica a través del flujo del diagrama de comportamiento, en nuestro caso, el diagrama de actividad. De forma ilustrativa, en la Figura 9 se muestran las posibles rutas (en amarillo) que tendría un diagrama de actividad, siendo cada una un “escenario de caso de uso”.

Por lo tanto, la generación de los escenarios y los diferentes casos de prueba se pueden obtener recorriendo todos los caminos posibles y, por

cada uno de estos caminos se identifican los pasos concretos de la solución global o de la funcionalidad específica del *smart contract*, incluidas las reglas y condiciones (tablas de decisión) establecidas, y se van añadiendo al caso de prueba en el mismo orden.

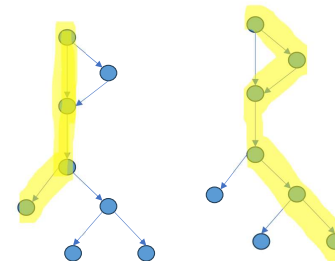


Figura 9. Ejemplo de caminos de escenarios de Caso de Uso

Una vez finalizado el proceso, se obtienen tantos casos de prueba como caminos se hubieran identificado en el requisito funcional —el que se representó como diagrama de actividad— y cada caso de prueba tendrá un comportamiento distinto, que coincidirá con el camino tomado y el juego de datos de prueba incluido en cada momento.

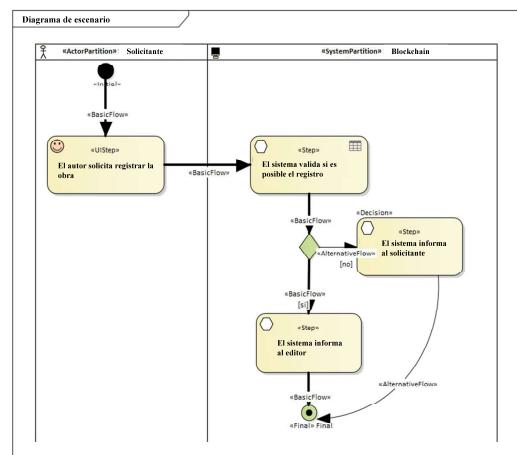


Figura 10. Diagrama de escenario

En la Figura 10, de forma ilustrativa, se muestra un diagrama de escenario obtenido a partir del diagrama de actividad de la Figura 7 que tiene en cuenta nuestra propuesta de metamodelo de pruebas funcionales.

Como paso final, en la Figura 11 se muestran los casos de prueba obtenidos a partir del diagrama de escenario anterior.

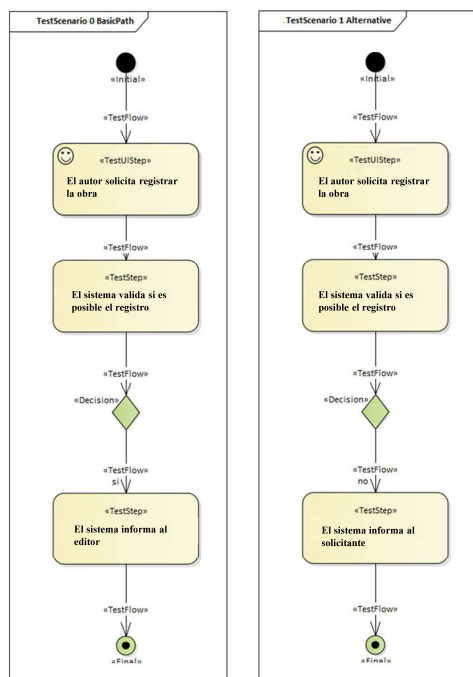


Figura 11. Diagramas de casos de prueba

4. Conclusiones y trabajos futuros

Las nuevas tecnologías que están apareciendo ofrecen un importante reto en todos los sectores y el mundo de las publicaciones digitales no es una excepción. La tecnología *blockchain* es una herramienta que permite la identificación unívoca de información (obras literarias, en el contexto de este artículo) y ofrece una solución para poder obtener una trazabilidad de cada uno de los ítems de una manera segura, eficiente y con unos costes asumibles por autores, editores y todos los actores implicados.

Sin embargo, el desarrollo bajo la tecnología *blockchain* requiere que exista una comunicación fluida entre usuarios y expertos funcionales durante todo el ciclo de vida de desarrollo, de tal manera sea posible especificar, verificar y validar los *smart contracts* (los cuales gobiernan las reglas de negocio del sistema) antes de su despliegue en una red *blockchain*. En este sentido, este artículo describe una propuesta basada en la ingeniería dirigida por modelos y orientada al usuario para obtener sistemáticamente pruebas funcionales a partir de la especificación de los *smart contracts*.

Esta propuesta ha sido validada dentro del proyecto SmartISBN, proyecto I+D+i de la empresa Lantia Publishing en colaboración con la Universidad de Sevilla. En este proyecto se ha

desarrollado una plataforma software que permite gestionar el proceso de edición, publicación y distribución de obras literarias, utilizando la tecnología *blockchain*. En concreto, la propuesta presentada ha permitido generar pruebas funcionales del sistema global y de los *smart contracts*, de forma sistemática, a partir de las especificaciones funcionales y de los acuerdos, reglas y condiciones acordadas. La puesta en marcha de la plataforma SmartISBN, gracias al uso de la tecnología *blockchain*, está permitiendo obtener una trazabilidad completa del proceso de edición, producción y distribución de obras literarias, gracias a las características intrínsecas de esta tecnología.

Como trabajos futuros planteamos mejorar nuestros protocolos de comunicación para poder generalizarlos y aplicarlos a otros entornos. De hecho, en la actualidad estamos trabajando en otro proyecto en el que se extiende nuestra propuesta y estas ideas a un contexto internacional.

En el contexto de las pruebas funcionales, nuestra idea es seguir mejorando estos mecanismos para indicar la priorización de las pruebas, no sólo para la generación.

La idea es que el equipo de trabajo no solo pueda generar las pruebas funcionales a partir de las especificaciones, si no que, además, los escenarios de caso de uso y los casos de prueba aparezcan priorizados para que, en caso de falta de recursos o tiempo, se priorice la generación de pruebas de funcionalidades más críticas.

Por otro lado, ya estamos trabajando en la generación automática del código a partir del modelo de *smart contract*. Se pretende utilizar los diagramas vistos anteriormente, donde se especifican los aspectos estructurales y de comportamiento del *smart contract* y, mediante transformaciones M2T (*Model2Text*) implementar el código de estos programas informáticos.

5. Agradecimientos

Este artículo ha sido elaborado gracias a los siguientes proyectos: EQUAVEL (PID2022-137646OB-C31), el cual es financiado por el Ministerio de Ciencia e Innovación del Gobierno de España; y el proyecto SmartISBN, el cual fue financiado por el CDTI, Entidad Pública Empresarial dependiente del Ministerio de Ciencia e Innovación.

Notas

- (1) SmartISBN es un proyecto financiado por el Centro para el Desarrollo Tecnológico Industrial (CDTI) y cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER). El objetivo de este proyecto es promover el desarrollo tecnológico, la innovación y una investigación de calidad que permita a editoriales, distribuidores, autores y agencias tener información instantánea sobre el funcionamiento

comercial de cualquier lanzamiento. Lantia Publishing, la empresa beneficiaria de la ayuda, quiere revolucionar el sector editorial a través de la *blockchain* y durante el presente proyecto ha desarrollado un prototipo no comercializable siguiendo las bases del organismo subvencionador. Actualmente, el proyecto está en fase de pilotaje, antes de su puesta en producción.

- (2) ONIX (Needleman, 2001) es un estándar abierto e internacional para la codificación y el intercambio electrónico de información bibliográfica y comercial en la industria editorial, con la participación de representantes de la cadena comercial editorial de más de veinte países (incluido España).
- (3) Debido a limitaciones de espacio, la descripción del metamodelo de *smart contracts* se focaliza en la descripción de sus metaclases, obviando los atributos concretos de las mismas.
- (4) El objetivo de usar el estándar DMN (Janssens et al., 2016) para la modelización de decisiones es poder trazar modelos con opciones complejas de forma sencilla y comprensible. Gracias a este enfoque, las partes implicadas pueden entender las decisiones que se toman, creando así más transparencia y claridad mediante el uso de modelos de decisión y el análisis resultante.

Referencias

- Alharby, M.; Aldweesh, A.; van Moorsel, A. (2018). *blockchain-based smart contracts: A systematic mapping study of academic research* (2018). // 2018 International Conference on Cloud Computing, Big Data and blockchain (ICCB) (pp. 1-6). IEEE.
- Bézivin, J. (2004). In search of a basic principle for model driven engineering. // *Novatica Journal*. 5:2, 21-24.
- Chirinos, C. P.; Furones, A. R. (2019). Blockchain: fundamentos técnicos y de negocio. // *Revista de Obras Públicas: Órgano profesional de los ingenieros de caminos, canales y puertos*. 3615, 42-47.
- Fontela, C. (2012). UML: modelado de software para profesionales. Alpha Editorial.
- Gramoli, V. (2022). *Blockchain Fundamentals*. // *Blockchain Scalability and its Foundations in Distributed Systems*. Cham: Springer International Publishing. 17-39.
- Jain, S.; Joshi, H. (2016). Impact of early testing on cost, reliability and release time. // 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). IEEE. 318-322.
- Janssens, L.; Bazhenova, E.; De Smedt, J.; Vanthienen, J.; Denecker, M. (2016). Consistent Integration of Decision (DMN) and Process (BPMN) Models. // *CAISE forum*. 1612, 121-128.
- Magadán-Díaz, M.; Rivas-García, J. I. (2020). La industria editorial española: dos décadas clave de transformación y cambio (1996-2016). // *Investigaciones de Historia Económica* (2020). <https://doi.org/10.33231/j.ihe.2020.04.003>
- Martínez Alés, R. (2001). *Información Comercial Española*. // *ICE: Revista de economía*. ISSN 0019-977X. 792, 109-123.
- Needleman, M. H. (2001). ONIX (online information exchange). // *Serials Review*. 27:3-4, 102-104.
- Safdar, M.; Qutab, S.; Ullah, F. S.; Siddique, N.; Khan, M. A. (2022). A mapping review of literature on Blockchain usage by libraries: Challenges and opportunities. // *Journal of Librarianship and Information Science*, 09610006221090225.
- Sánchez-Gómez, N.; Torres-Valderrama, J.; García-García, J. A.; Gutiérrez, J. J.; Escalona, M. J. (2020). Model-based software design and testing in blockchain smart contracts: A systematic literature review. // *IEEE Access*. 8, 164556-164569.
- The Medizine (2021). El blockchain llega al mundo editorial de la mano de Lantia Publishing. <https://themedizine.com/p/el-blockchain-llega-al-mundo-editorial-de-la-mano-de-lantia-publishing> (2023-06)
- Yuan, X.; Zhou, S. (2022). Research on Library Intelligent Service Based on Blockchain Technology. // *International Conference on Computer Science and Education*. Singapore: Springer Nature Singapore. 267-277.
- Zeng, J.; Dai, X.; Xiao, J.; Yang, W.; Hao, W.; Jin, H. (2019). Bookchain: Library-free book sharing based on Blockchain technology. // 2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN). IEEE. 224-229.
- Zou, W.; Lo, D.; Kochhar, P. S.; Le, X. B. D.; Xia, X.; Feng, Y.; [...] ; Xu, B. (2019). Smart contract development: Challenges and opportunities. // *IEEE Transactions on Software Engineering*. 47:10, 2084-2106.

Enviado: 2023-03-30. Segunda versión: 2023-05-05.
Aceptado: 2023-11-02.

