Ingeniería en Software - Trabajo Final de Graduación (PIA)

Hacia una Ingeniería en Software orientada a *Blockchain*.

Propuesta de un modelo de desarrollo de software basado en el análisis de la tecnología y las prácticas de desarrolladores Blockchain en Argentina.

Autor: Zamora Fernando Gabriel

Legajo: SOF00758

Tutora: Ana Carolina Ferreyra

2019

Abstract

Blockchain technology is known primarily as responsible for current cryptocurrency systems. Since its conception as a database or set of transactional records of a shared, distributed and secure nature, the Blockchain has gained great relevance, not only in the world of computing but also in the business and industrial field. The technical maturation and in-depth knowledge of its principles, made this technology is extrapolated to the use cases that exceed the financial sector for which it was originally devised. The incorporation of programs that can be executed in it, called Smart Contracts, improves its scope allowing redefining multiple processes, making technology the candidate par excellence for those systems that require trust, security, transparency, efficiency and reduction of operating costs. There are so many benefits and potential attributed to this technology, that it is sometimes compared to the beginning of the Internet and the web revolution.

The increase in popularity around Blockchain technology quickly manifested itself in an exponential growth of developments that incorporate it as a central component in their proposal. This kind of race for innovation, showed the use in many cases excessive and unnecessary of this technology. However, the critical point in this context is the absence of good software engineering practices during development, which translates directly into products resulting from low quality due to unpredictable, poorly controlled and reactive processes.

This Final Graduation Work seeks to investigate and respond to this problem, proposing a process model for the development of Blockchain-oriented software that

allows managing and developing a project in an orderly manner using tools and principles that lead to quality results. For this, the proposed model draws on the analysis of technological principles, the trends currently used in the development oriented to Blockchain and Smart Contracts, and the contributions of other authors on this field.

Keywords: Software Engineering, Blockchain, Smart Contracts, Software Process Model.

Resumen

La tecnología *Blockchain* o Cadena de Bloques en su traducción al español, es conocida principalmente por posibilitar y ser el sostén de los actuales sistemas de criptomonedas. Desde su concepción como una base de datos o conjunto de registros transaccionales de carácter compartido, distribuido y seguro, *Blockchain* ha ganado gran relevancia, no sólo en el mundo de la computación sino también en el ámbito empresarial e industrial. La maduración tecnológica y el conocimiento en profundidad de sus principios, permitió que Blockchain sea extrapolada a casos de uso que exceden el sector financiero para el cual fue originalmente ideada. La incorporación de programas ejecutados en ella, llamados *Smart Contracts* o Contratos Inteligentes, potenciaron su alcance haciendo posible redefinir múltiples procesos, convirtiendo a esta tecnología en la candidata por excelencia para aquellos sistemas que precisan de confianza, seguridad, transparencia, eficiencia y reducción de costos operativos. Son tantos los beneficios y el potencial que se le atribuyen, que algunos llegan a compararla con la revolución de internet y la web en sus inicios.

El aumento en la popularidad alrededor de *Blockchain* rápidamente se manifestó en un crecimiento exponencial de desarrollos que la incorporan como un componente central en su propuesta. Esta especie de carrera por la innovación, dejó entrever la utilización, en muchos casos desmedida e innecesaria, de esta tecnología. Sin embargo, el punto crítico en este contexto, es la ausencia de buenas prácticas de la ingeniería en software durante el desarrollo, lo cual se traduce directamente en productos resultantes de baja calidad, fruto de procesos impredecibles, mal controlados y reactivos.

El presente Trabajo Final de Graduación busca indagar y dar respuesta a este problema, proponiendo un modelo de procesos para el desarrollo de software orientado a *Blockchain* que permita gestionar y desarrollar un proyecto de manera ordenada, valiéndose de herramientas y principios que conduzcan a resultados de calidad. Para ello el modelo propuesto se nutre del análisis de los principios tecnológicos, las tendencias empleadas actualmente en el desarrollo orientado a *Blockchain* y *Smart Contracts*, y los aportes de otros autores sobre este campo.

Palabras clave: Ingeniería en Software, Blockchain, Smart Contracts, Modelo de Proceso de Software.

Tabla de contenido

Lista de tablas

Lista de figuras

Introducción

Blockchain desde su aparición en el año 2008 a través de la publicación de Satoshi Nakamoto ha demostrado un interés creciente a lo largo de la última década.

Si bien la primera implementación de la Cadena de Bloques fue Bitcoin (2009), un sistema de dinero electrónico posibilitador de transacciones sin organismos financieros intermediarios (Nakamoto, 2008), los casos de uso de *Blockchain* se extendieron a diversas industrias debido a su potencial y el desarrollo de nuevos protocolos informáticos vinculados como los *Smart Contracts*[1]. Estos avances y maduración de la tecnología han conseguido que *Blockchain* trascienda el plano de las criptomonedas para cautivar a grandes organizaciones y startups alrededor del mundo. La tecnología *Blockchain* representa un cambio de paradigma, evidenciando en sus cualidades, un gran potencial para la generación de nuevos modelos de negocio y el mejoramiento de procesos sistemáticos existentes en la actualidad bajo enfoques centralizados.

En esencia *Blockchain*, es un conjunto de tecnologías que permiten mantener un registro distribuido, descentralizado, sincronizado y muy seguro de la información; se accede a través ordenadores y otros dispositivos conectados a internet, cumpliendo la función de un libro mayor de transacciones digitales, permitiendo identificar las partes involucradas, almacenar y trazar esa información en todo momento.

[1] Un contrato inteligente (en inglés *Smart contract*) es un programa informático que facilita, asegura, hace cumplir y ejecuta automáticamente acuerdos registrados entre dos o más partes.

La encuesta llevada a cabo por la consultora PwC[2] sobre la tecnología *Blockchain* indica que el 84% de los 600 ejecutivos pertenecientes a 15 territorios alrededor del mundo, se encuentran activamente involucrados en proyectos que la contemplan (PwC, 2018). Por su parte la tendencia a apostar a esta tecnología, también se ve reflejada por las inversiones de tipo *Venture Capital* (VC) en startups *Blockchain;* según datos provistos por la compañía PitchBook especializada en el análisis de datos de capitales privados, citados en un artículo por Diar[3], solo en los tres primeros trimestres de 2018 las compañías de *Blockchain* y cryptomonedas han recaudado casi 3.9 mil millones de dólares a través del VC tradicional, un 280% más en comparación con el año 2017 (Diar Ltd , 2018).

No obstante, desde la mirada de la Ingeniería en Software y en acuerdo con lo expuesto por Porru, Pinna, Michele, & Tonelli (2017) el crecimiento acelerado en el número de proyectos de software que nacen y se desarrollan en torno a las diversas implementaciones de *Blockchain* deja entrever un proceso de software apresurado y no regido. El escenario se percibe como una suerte de carrera, donde las empresas y sobre todo las startups compiten por el lanzamiento temprano de software, sin garantizar la calidad, ni teniendo en cuenta todos los conceptos esenciales de la ingeniería.

La falta de buenas prácticas durante el desarrollo, quedó expuesta a través de los ciberataques que tuvieron como objetivo a esta tecnología y el software asociado a ella.

[2] PwC (abreviatura de PriceWaterhouseCoopers) es reconocida como una de las firmas de consultoría dentro de las Big Four, junto con Deloitte, KPMG y EY.

[3] Diar es una firma que proporciona cobertura concisa y análisis de desarrollos significativos dentro de la industria global de criptomonedas.

Acorde a un artículo publicado por el MIT Technology Review, los hackers han robado casi 2 mil millones de dólares en criptomonedas en lo que va desde principios de 2017 hasta la actualidad. Los vectores de ataque identificados fueron diversos, puntualmente se reconocen los perpetrados a la infraestructura *Blockchain,* como el ataque del 51% que sufrió Ethereum Clasic; los ataques a software cliente, como a billeteras y casas de cambio*;* y por último, de gran preocupación en la comunidad *Blockchain* actual, los ataques a través de vulnerabilidades en contratos inteligentes (Orcutt, 2019).

Partiendo del contexto expuesto con anterioridad, el presente Trabajo Final de Graduación, situado desde la experticia de la disciplina que lo enmarca, ofrece un análisis de la tecnología *Blockchain,* el ecosistema de desarrolladores argentinos y se vale de propuestas generadas por investigaciones anteriores para arribar a la proposición de un modelo de proceso de desarrollo de software orientado a *Blockchain.*

Para cumplir con tal fin, posterior a la definición metodológica que guió el desarrollo, el trabajo se organiza en 4 capítulos: el capítulo 1, parte del análisis de la tecnología *Blockchain* y *Smart Contracts*; el capítulo 2, examina las prácticas llevadas adelante por los desarrolladores *Blockchain* argentinos mediante un estudio de campo; el capítulo 3, se enfoca en analizar los trabajos realizados sobre áreas de conocimiento relativas a la ingeniería de software aplicados a esta tecnología, con el fin de recuperar contribuciones de otros autores, útiles para el presente Trabajo Final de Graduación; el capítulo 4, responde a la aplicación de los conocimientos recabados en los apartados anteriores, en él se propone un modelo de proceso de desarrollo de software orientado a

Blockchain (MDSOB); por último se presentan las conclusiones finales e ideas futuras producto de la investigación realizada.

Delimitación del problema

La disciplina de Ingeniería en Software se formaliza en 1968, en una conferencia realizada para discutir lo que en aquel tiempo se llamaba la crisis del software. En ese entonces y años posteriores, se volvió notorio que los enfoques individuales al desarrollo de programas informáticos no escalaban hacia los grandes y complejos sistemas de software requeridos por las empresas. Éstos no eran confiables, costaban más de lo esperado y se distribuían con demora. Durante esta crisis que duró prácticamente 3 décadas (60´s, 70´s y 80´s), fueron diversos los autores que trataron el tema; en 1986 Fred Brooks argumenta en su artículo científico *No silver bullet*, que por la propia naturaleza del software resulta imposible pensar en un desarrollo tecnológico o técnica de gestión, que por sí solo prometa una mejora en la productividad, fiabilidad, y simplicidad relativa a la construcción de este tipo de productos (Brooks, 1987). Por su parte, en 1994, se publica un artículo de W. Wayt Gibb titulado *Software's Chronic Crisis* donde el autor, basándose en hechos reales, presenta la idea de que a pesar de los progresos, la disciplina llamada Ingeniería en Software seguía siendo una aspiración; Wayt Gibb expone en su artículo que la mayor parte del código de computadoras que se confeccionaba hasta el momento era realizado a mano por artesanos que utilizaban lenguajes de programación bastante burdos y técnicas que no se podían medir ni repetir con consistencia (Gibbs, 1994).

En la actualidad, si bien la disciplina de la Ingeniería en Software ha madurado exponencialmente dejando atrás lenguajes rústicos y proponiendo metodologías que permiten un desarrollo más controlado y orientado a la calidad, cuando se habla puntualmente de *Blockchain*, se reconoce un cambio de paradigma en la construcción de software. Este cambio requiere nuevas formas de desenvolverse durante la actividad profesional, por lo tanto, determinadas prácticas que hasta el momento se consideraban una suerte de bala de plata, deben ser reevaluadas dentro de este nuevo tejido tecnológico.

El fracaso de los proyectos *Blockchain* es una realidad, según la Academia China de Tecnología de la Información y las Comunicaciones (CAICT), solo el ocho por ciento de los más de 80,000 proyectos *Blockchain* que se lanzaron globalmente todavía está activo en la actualidad. Los proyectos *Blockchain* sólo promedian una vida útil de aproximadamente 1,22 años según el reporte emitido por dicha institución en el marco de la International Big Data Industry Expo 2018 en China. Al respecto, Dante Disparte, un especialista en riesgos, analiza los factores de fracaso en un artículo para la revista Forbes (2019), mencionando y posicionando como uno de los principales causantes a la falta de madurez que envuelva a la tecnología y la falta de higiene cibernética en los proyectos *Blockchain*.

Basado en el panorama descripto con anterioridad y retomando las ideas de lo que fue la llamada crisis del software, es posible percibir una coyuntura similar a la de ese entonces; la comunidad avocada a la tecnología *Blockchain* parece precipitarse sobre la

misma, actuando de manera cuasi artesanal, dejando de lado prácticas ingenieriles, a cambio de una pronta inserción en el mercado de sus productos de software.

Justificación e importancia de la investigación

Como se introdujo en la delimitación del problema, una situación similar a la famosa crisis de software empezó a manifestarse en la comunidad de desarrolladores *Blockchain.* Iniciativas como las conferencias realizadas y apañadas por la IEEE[4], las diferentes asociaciones, fundaciones y comunidades que se fueron consolidando desde su aparición, o inclusive la propuesta de una estandarización en marcha a través de la ISO/TC 307[5], denotan la necesidad inminente de fijar estándares que permitan el progreso de la tecnología *Blockchain*.

El esfuerzo invertido en el presente Trabajo Final de Graduación se ve justificado por diversos ejes; i) recabar información técnica sobre esta tecnología y ponerla a disposición de los interesados, significa brindar aportes teóricos que pueden convertirse en un marco de referencia para futuras investigaciones e implementaciones en el desarrollo de sistemas que involucren la Cadena de Bloques como uno de sus componentes; ii) analizar las prácticas llevadas adelante por los desarrolladores *Blockchain* en la Argentina, permite obtener una mirada objetiva acerca del ecosistema

[4]El *Institute of Electrical and Electronics Engineers* es una rreconocida asociación mundial de ingenieros dedicada a la normalización y el desarrollo en áreas técnicas. https://*Blockchain*.ieee.org/conferences.

[5] ISO/TC 307 es una norma internacional promovida por ISO (Organización Internacional de Normalización) que busca estandarizar las tecnologías Blockchain y de Registros Distribuidos en general.

local y en cierta forma validar la necesidad real de un modelo de proceso de desarrollo de software; iii) investigar y analizar las propuestas generadas en el campo de la Ingeniería en Software orientada a *Blockchain* durante el periodo comprendido entre los años 2017 y 2018, constituye en sí el estado de arte del enfoque Ingenieril sobre dicha tecnología, del cual es posible recuperar herramientas, técnicas y modelos que sean aplicables a diferentes etapas del ciclo de vida del software. Esta sección tiene un doble propósito, por un lado, de cara al presente trabajo, dará sustento y proveerá información que nutra el modelo propuesto como motivación final de la investigación, y por el otro, de cara al lector, despliega un abanico de posibilidades a ser exploradas y aplicadas al ejercicio del desarrollo de software *Blockchain*; iv) finalmente, un abordaje integral mediante la propuesta de un modelo de proceso de desarrollo de software orientado a *Blockchain*, brinda una visión macro de cómo abordar las necesidades, el análisis y diseño inherentes a un proyecto fundado en esta tecnología proponiendo un camino sistematizado factible de aplicación durante el ciclo de vida de desarrollo.

Desde una mirada personal el trabajo se evidencia como una herramienta para la divulgación de conocimiento relativo a la tecnología *Blockchain*; es implícito el ánimo de reforzar e impulsar mediante el presente Trabajo Final de Graduación la necesidad de una visión ingenieril sobre esta tecnología, haciendo un llamado a los profesionales y practicantes de las Ciencias de la Computación a reflexionar y realizar sus aportes pertinentes, que alimenten la práctica y cuerpo de conocimiento concerniente a *Blockchain*.

Preguntas de investigación

Partiendo de la identificación del problema y los motivos por los cuales es de importancia llevar adelante este Trabajo Final de Graduación, se plantean una serie de preguntas que guían el diseño de la investigación, y permiten definir métodos, procedimientos y herramientas a ser utilizadas. A su vez, las preguntas se constituyen, en sí mismas, como delimitadoras y orientadoras de la investigación en función de reducir al máximo las posibles ambigüedades que puedan presentarse sobre el abordaje de la temática objeto.

- ¿Cómo funciona la tecnología *Blockchain* y *Smart Contracts*?
- ¿Cuáles son las prácticas empleadas por la comunidad argentina para el desarrollo de software *Blockchain*?
- ¿Qué avances e investigaciones existen sobre métodos, técnicas y herramientas para el desarrollo *Blockchain* desde una perspectiva ingenieril?
- ¿Qué enfoque de la Ingeniería en Software se ajusta mejor a las características propias del desarrollo sobre la tecnología *Blockchain*?
- ¿Cómo puede modelarse un proceso que mitigue el fracaso de un producto de software implementado en la tecnología *Blockchain*?

Objetivos de la investigación

Los objetivos perseguidos por el presente Trabajo Final de Graduación se definen partiendo de las preguntas de investigación enunciadas con anterioridad. Este proceder

permite generar coherencia y un fuerte eje que articule los intereses del proyecto desde sus inicios.

Objetivo general.

- Diseñar y proponer un modelo de procesos para el desarrollo de software sobre la tecnología *Blockchain,* fundado en el análisis tecnológico subyacente, la identificación de prácticas empleadas por desarrolladores *Blockchain* en Argentina y aportes recuperados de anteriores investigaciones.

Objetivos específicos.

- Establecer una visión general sobre el funcionamiento de la tecnología *Blockchain* y *Smart Contracts*.
- Identificar los diferentes principios que subyacen a la tecnología *Blockchain*.
- Recabar y examinar las prácticas llevadas adelante en el desarrollo de aplicaciones *Blockchain* por la comunidad argentina.
- Examinar y evaluar los aportes realizados por otros autores a través de artículos científicos en el ámbito de simposios y congresos destinados al abordaje de la Ingeniería de Software orientada a *Blockchain*.
- Analizar la importancia y los principios vinculados a los procesos y modelos de proceso de software y cómo pueden beneficiar el desarrollo orientado a *Blockchain*.

Marco metodológico

En el presente apartado se especifica el tipo de investigación, las técnicas y los instrumentos utilizados para conseguir los objetivos planteados para el trabajo.

Resulta interesante y esclarecedor como primera aproximación a la definición de una metodología, considerar el artículo realizado por Celso De La Cruz Casaño titulado Metodología de la Investigación Tecnológica en Ingeniería (2016); en el mencionado trabajo, el autor citando a Garcia (2012), diferencia al paradigma tecnológico del paradigma clásico (cualitativo, cuantitativo); en el primer caso el autor sostiene que el foco esta puesto en transformar la realidad, mientras que en el segundo, la perspectiva esta puesta en darle una explicación o comprenderla. A pesar de esta distinción, Casaño aclara que es sumamente necesario el estudio y el conocimiento de la realidad para establecer los cimientos sobre los que se apoya la labor tecnológica (De La Cruz Casaño, 2016). Esta idea de fundarse en el conocimiento para tomar acción, es también sostenida en la taxonomía que diferencia a la investigación básica de la investigación aplicada, siendo en el caso de la última, la que compete a este trabajo, necesario construir una base teórica para la resolución puntual del problema.

Partiendo de este enfoque, el diseño metodológico empleado para desarrollar el presente Trabajo Final de Graduación en el ámbito de la carrera de Ingeniería en Software, muestra una división lógica de dos partes que se complementan y cumplen con la manera de proceder enunciada con anterioridad. Además, en lo que respecta a las bases teóricas y de conocimiento necesarias para la elaboración de la propuesta, son abordadas desde una aproximación documental y de campo.

La realización de la investigación se ve definida y guiada por el siguiente proceso, donde el orden de las etapas garantiza un desarrollo sistematizado del trabajo de recopilación, análisis y formulación de una propuesta.

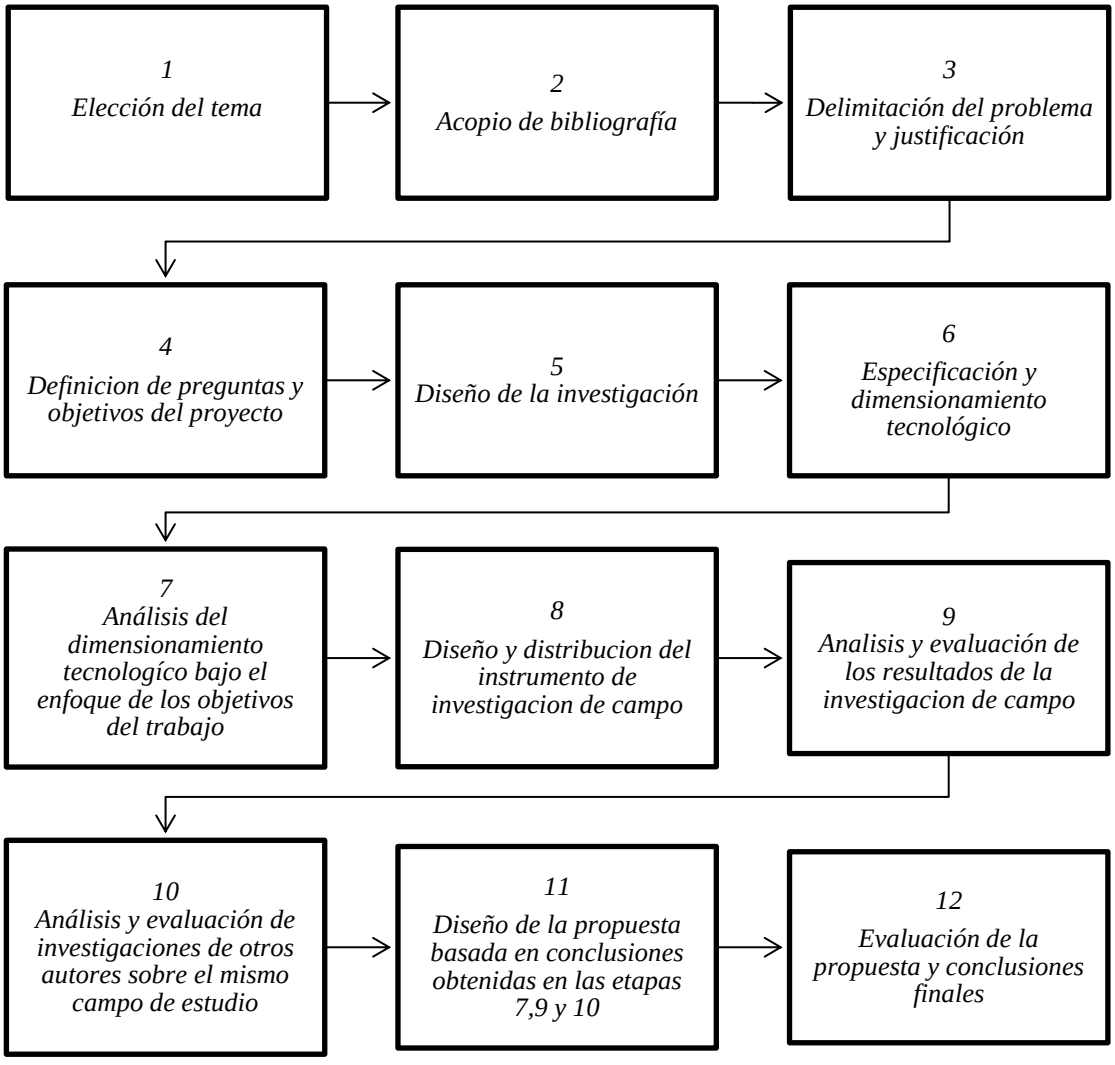

Figura 1. Diagrama de la metodología de investigación aplicada al proyecto.

Diseño de la investigación documental

En lo que respecta a la indagación técnica y conceptual de la tecnología *Blockchain*, *Smart Contract* y los avances en la Ingeniería en Software orientada a *Blockchain*, el diseño de investigación responde al tipo documental. Este tipo de diseño "es un proceso basado en la búsqueda, recuperación, análisis, crítica e interpretación de datos secundarios, es decir, los obtenidos y registrados por otros investigadores en fuentes documentales: impresas, audiovisuales o electrónicas. Como en toda investigación, el propósito de este diseño es el aporte de nuevos conocimientos" (Arias Odón, 2012).

A pesar de que se habla de fuentes que aportan datos secundarios, en este tipo de investigación, es posible discriminarlas, a su vez, en primarias, cuando se trata de obras originales, y secundarias, cuando se emplea material en el que se hace referencia a la obra de un autor (Arias Odón, 2012). En el caso del presente trabajo, se emplean ambos tipos de fuentes, siendo el soporte en su totalidad electrónico, tanto de artículos científicos como de libros digitalizados.

Es importante remarcar, además, que acorde al objeto de estudio, resulta de suma utilidad consultar sitios webs y blogs, que si bien, teóricamente su valor como fuente no es elevado, es imprescindible tenerlos en cuenta frente a una temática tan reciente y de avances vertiginosos; los blogs y sitios web significan en este aspecto un medio para mantenerse actualizado con inmediatez sobre las novedades que envuelven la tecnología *Blockchain*.

Para llevar adelante la tarea de recolección de material bibliográfico/ hemerográfico, es indispensable el uso de motores de búsqueda de artículos indexados disponibles en Internet o World Wide Web; los principales utilizados en este trabajo son Google y sus derivados, Google Books y Google Academic, así como también ResearchGate[6] y arXiv.org[7]. Otro sitio relevante para el acceso a información centrado en la temática *Blockchain* es *Blockchain* Library[8], así como los repositorios de Github decrypto-org/*Blockchain*-papers y bellaj/*Blockchain*

Por otro lado, otra técnica empleada fue el mapeo de las referencias en aquellas fuentes documentales de carácter secundario a través de su *doi* (Identificador de objeto digital).

En cuanto a los instrumentos, se emplean fichas bibliográficas/ hemerográficas, y fichas de contenido, todo bajo el entorno de trabajo proporcionado por el software libre de gestión bibliográfica Zotero. Las herramientas proporcionadas por esta plataforma son de gran utilidad y permiten además de la generación de las correspondientes fichas, organizar el material en colecciones, incluir anotaciones, asignar palabras claves y relacionar referencias entre sí; por otro lado, simplifica la tarea de búsqueda permitiendo visualizar las referencias acordes a palabras claves o campos definidos en las fichas.

---

[6] Red global que reúne profesionales de la ciencia y la investigación donde se comparten estudios de gran valor académico.

[7] Servidor de almacenamiento y distribución de archivos científicos administrado por la Cornell University.

[8] Sitio web cuya misión es proporcionar una biblioteca digital y un archivo de recursos en *Blockchain* y Criptomonedas para investigadores de todo el mundo.

Como en su mayoría los datos recabados para estas secciones son de carácter cualitativo, la forma de procesarlos es a través de técnicas de análisis-síntesis; se parte de la lectura minuciosa de los textos y la recuperación de las ideas principales a través de resúmenes, por otro lado, se emplean técnicas como la categorización de datos y el desarrollo de tablas comparativas que permiten visualizar información relevante mediante su confrontación. Además, de manera implícita, se usaron mapas conceptuales para organizar el contenido de manera coherente previo y durante el desarrollo del trabajo de investigación.

Diseño de la investigación de campo

En lo que respecta al estudio de campo sobre al análisis de las prácticas de la comunidad de desarrolladores *Blockchain* en Argentina, se utiliza como instrumento la encuesta online, éste permite, mediante el posterior procesamiento de los datos recolectados, explorar y evaluar la situación.

La elección de una encuesta como instrumento, es simple, a través de esta herramienta y sus características como la velocidad de respuesta, la atemporalidad y ubicuidad, es posible abarcar una muestra de mayor tamaño para su distribución, que en este caso corresponde a los desarrolladores de software avocados a esta tecnología en el territorio argentino; se estipula una participación mínima de 5 individuos, número prudente considerando lo reciente de esta tecnología, así como los diferentes roles y perspectivas dentro de un proyecto de software.

Para su confección se emplea la conocida herramienta Google Forms, hacerlo por este medio permite el acopio ordenado de datos a la vez que facilita la distribución de las preguntas, recepción de las respuestas y posterior análisis de los resultados.

Dentro del cuestionario, se incorporan preguntas tanto cerradas como abiertas; a su vez, este instrumento, se divide en dos secciones, una con el objetivo de recabar datos demográficos y otra que corresponde a la indagación de las prácticas empleadas a lo largo del ciclo de vida de desarrollo de software; para esto último se consideraron las principales áreas del conocimiento según el SWEBOK o las fases genéricas del ciclo de vida del software (requerimientos, diseño, construcción, pruebas y mantenimiento).

Partiendo del objetivo específico del proyecto "Recabar y examinar las prácticas llevadas adelante en el desarrollo de aplicaciones *Blockchain* por la comunidad argentina", la encuesta desarrollada se basa ampliamente en la utilizada por Garousi, Coşkunçay, Betin-Can, y Demirörs en su artículo A Survey of Software Engineering Practices in Turkey; donde los autores presentan un diseño del instrumento de indagación basado en el modelo GQM (Goal/ Question/Metric). La pertinencia, relativa contemporaneidad (año 2014), completitud y efectividad del instrumento mencionado, encaja casi a la perfección con la esencia de la pesquisa que se persigue en el presente proyecto. Por tal motivo, se retoman muchas de las preguntas empleadas en el mismo, realizando los ajustes necesarios para extrapolar el análisis al territorio argentino y puntualmente al desarrollo *Blockchain*.

El proceso de elaboración y análisis de este instrumento esta dado por las siguientes etapas:

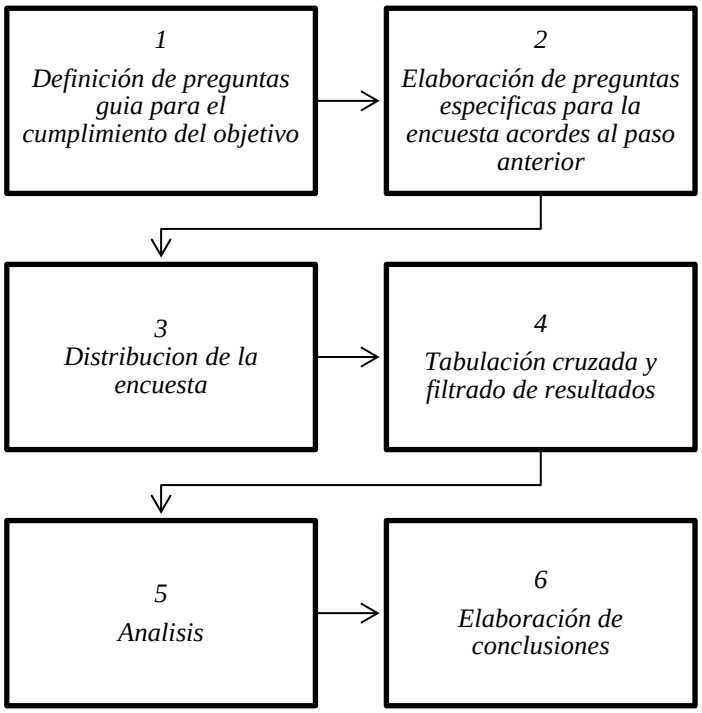

Figura 2. Proceso para la elaboración e interpretación de los resultados de la encuesta.

La encuesta será distribuida en grupos cerrados de la red social Facebook que fueron recomendados por la asociación Bitcoin Córdoba y monitoreados durante 5 meses previos a la publicación del cuestionario. En dichos grupos se manifiesta constante actividad, contenido de calidad y reúne miembros de toda la Argentina, así como referentes en la materia. Estos grupos son: Programación Bitcoin & *Blockchain*, Bitcoin Argentina y Córdoba Bitcoin.

Grilla de investigación

Para concluir este apartado, se presenta a continuación una grilla de investigación que sintetiza la metodología empleada. Es importante aclarar que, por el tipo de

investigación y la metodología dominante en ella, existe cierta flexibilidad en el abordaje del trabajo que puede exceder lo expuesto a continuación, por lo cual, los elementos detallados en la grilla son meramente a modo orientativo.

Tipo de investigación	Investigación aplicada
Metodología	Cualitativa Inductiva, Documental
Técnicas de investigación	Análisis-Síntesis, Encuesta
Instrumento	• Fichas Bibliográficas/Hemerográficas y de contenido • Tablas comparativas • Cuestionario con preguntas abiertas y cerradas
Población/ Corpus de análisis	• Artículos académicos y científicos. Bibliografía especializada • Desarrolladores *Blockchain*
Muestra/ Recorte del corpus	• Temática asociada: *Blockchain*, *Smart Contracts*, Ingeniería de Software • Residentes en argentina (al menos 15 individuos)
Criterio muestral	• Pertinencia, actualidad y exhaustividad. • Muestreo no probabilístico (bola de nieve)

Figura 3. Grilla de investigación.

Planificación tentativa de actividades

Acorde al proceso definido para la investigación se planifico el desarrollo del trabajo de manera tentativa distribuido en meses y semanas, fijando el inicio en el mes de agosto del año 2018, con el objetivo de finalizarlo a fines del mes de marzo de 2019.

La granularidad de las actividades se mantiene a un nivel macro, esto es debido al diseño propuesto para la investigación el cual permite la flexibilidad en los plazos durante la ejecución del estudio y la posible necesidad de realizar tareas auxiliares no contempladas desde los inicios del proyecto.

Actividad	**AGO**				**SEP**				**OCT**				**NOV**				**DIC**				**ENE**				**FEB**				**MAR**			
Elección del tema																																
Acopio de bibliografía																																
Fichaje de fuentes																																
Delimitación del problema y justificación																																
Definición de preguntas y objetivos del proyecto																																
Diseño de la investigación																																
Especificación y dimensionamiento tecnológico																																
Diseño y distribución de la encuesta																																
Análisis y evaluación de los resultados																																
Análisis de propuestas de otros autores																																
Diseño del modelo de procesos																																
Redacción de conclusiones																																

Capítulo 1.

Tecnología *Blockchain* y de *Smart Contracts*

El presente capítulo tiene como objetivo el análisis en profundidad de la tecnología *Blockchain* y *Smart Contracts*.

Es fundamental para el desarrollo y comprensión de este Trabajo Final de Graduación, partir del entendimiento de dichas tecnologías, para tal fin se presenta un abordaje que abarca desde sus orígenes hasta los componentes esenciales que hacen a su funcionamiento, manteniendo en la medida de lo posible un enfoque abstraído de cualquier implementación particular.

1.1. El origen de la tecnología *Blockchain*

Las ideas fundamentales detrás de la tecnología *Blockchain* tienen sus orígenes a finales de los años 80, principios de los 90. En 1989, Leslie Lamport desarrolló e introdujo el protocolo Paxos, pero no fue hasta 1998 que la revista ACM Transactions on Computer Systems lo publicara en una de sus ediciones; en este escrito Lamport utiliza la metáfora de la civilización llamada Paxos para ilustrar el algoritmo que permitía conseguir el consenso dentro de una red, a pesar de la presencia de nodos defectuosos, mediante el acuerdo entre la mayoría. Lamport exhibe en su artículo un protocolo que proporciona una nueva forma de implementar el enfoque de máquina de estados para el diseño de sistemas distribuidos. Con Paxos, a los nodos de una red se les asignan roles vinculados a las funciones que desempeñan, estos son: proponente, aceptador, aprendiz y líder, quienes actuando con sinergia consiguen definir el estado dentro la red. Por otro

lado, el protocolo presentado por Lamport, incluye lo que denominó réplicas, con lo cual se refiere al estado común que se almacena en cada uno de los nodos (Lamport, 1998).

Contemporáneo al trabajo de Lamport, en 1991 Stuart Haber y W. Scott Stornetta desarrollan el concepto de una red de bloques protegida criptográficamente para brindar una solución computacionalmente práctica contra la manipulación de documentos digitales y su sello de tiempo. Su propuesta tenía el objetivo de conseguir la inmutabilidad de dichos documentos posterior a su almacenamiento (Haber & Stornetta, 1991). En 1992 Stuart Haber et al. actualizan su sistema incorporando los árboles de Merkle[9] en sus bloques, lo que mejoró la eficiencia permitiendo recopilar más documentos en cada uno de ellos.

Gracias a estos antecedentes, el 1 de noviembre del año 2008 sucedió el hito más significativo para la concepción de la tecnología *Blockchain* como se conoce en la actualidad; a través de una lista de distribución de emails[10], se dio a conocer un artículo firmado por Satoshi Nakamoto titulado Bitcoin: A Peer-to-Peer Electronic Cash System, en el mismo el autor retomaba las ideas de consenso, replica, Cadena de Bloques y protección criptográfica. En el documento, Satoshi presenta la tecnología resultante de vincular estos componentes mediante la aplicación en un sistema distribuido posibilitador de transacciones financieras sin la necesidad de agentes intermediarios, es decir prescindiendo de la confianza entre las partes involucradas (Nakamoto, 2008). Su implementación práctica demoró unos meses más en concretarse; el 4 de enero del año

[9] Merkle, Ralph. (1989). A Certified Digital Signature. 435. 218-238. 10.1007/0-387-34805-0_21.

[10] https://bit.ly/2SM1u5q

2009, también de la mano de su creador, se registró el primer bloque conocido como Bloque Genesis, dando nacimiento a la red Bitcoin, cuya tecnología subyacente es lo que se conoce como *Blockchain*.

1.2. Fundamentos de la tecnología *Blockchain*

Si bien no existe una definición formalmente estandarizada sobre la tecnología *Blockchain*, todos los autores que la refieren coinciden en ciertos aspectos que la describen y configuran como tal, partiendo de ello se propone el siguiente abordaje.

Una Cadena de Bloques o *Blockchain* es, en esencia, una base de datos distribuida en una red peer to peer. Cada transacción realizada en ella es verificada por los participantes de la red y aceptada a través de mecanismos de consenso, las mismas se organizan y almacenan en bloques matemáticamente relacionados entre sí, siendo la única función permitida la adición de nuevos registros. La información contenida en la *Blockchain* se encuentra criptográficamente protegida y es inmutable.

Para ahondar en esta definición y facilitar su comprensión, se analiza la misma descomponiéndola, como se introdujo hace momentos, en sus conceptos claves, los cuales dan cuenta de cómo aporta cada uno de ellos a esta tecnología.

1.2.1. Base de datos distribuida.

Al iniciar la definición elaborada anteriormente encontramos este primer concepto que demuestra la funcionalidad principal de la tecnología *Blockchain*; el mismo hace referencia a la capacidad de mantener una copia idéntica de un conjunto de datos,

almacenados de manera sistémica para su uso, en diferentes espacios lógicos y geográficos interconectados por una red de comunicaciones.

1.2.2. Peer to peer.

La tecnología de Cadena de Bloque presenta este modelo de comunicación donde cada nodo tiene relación directa con sus semejantes dentro de la red. Esto significa que no existe un control centralizado en las comunicaciones y es el componente motivador en los casos de uso donde se busca eliminar una tercera parte involucrada en la ejecución de transacciones.

En este tipo de comunicación, todos o algunos de los aspectos del sistema funcionan sin clientes ni servidores fijos, en su lugar, todos los nodos constituyentes de la red cumplen con ambos roles de manera simultánea respecto a las peticiones que lo mantienen vinculado al resto de los participantes.

1.2.3. Sólo adición.

Otra propiedad que se menciona es la de sólo adición. Esta propiedad establece el comportamiento de esta tecnología, la cual sólo permite a los nodos de la red agregar nuevos registros, los que se almacenan de manera secuencial y ordenados en el tiempo. En consecuencia, los datos registrados son inmutables, siendo imposible modificarlos o eliminarlos una vez agregados al conjunto de registros pertenecientes a la *Blockchain*.

1.2.4. Actualizable vía consenso.

Cada nueva transacción que se quiera registrar en una *Blockchain*, está sometida a un mecanismo de consenso. El objetivo de este, es posibilitar que todos los nodos que forman parte de la red lleguen a un acuerdo sobre el estado final de los datos almacenados, evitando de esta forma la adición de datos erróneos a la cadena. El consenso es un mecanismo que se logra a través de criterios estricta y claramente definidos en el protocolo de implementación de esta tecnología. Este principio es considerado uno de los componentes fundamentales y más importantes, ya que nutre de valor a esta tecnología permitiendo una verdadera descentralización y la conservación de la integridad de los datos.

1.2.5. Criptográficamente protegida.

Por último, y de gran valor en *Blockchain*, se encuentra el uso de la criptografía. En el ámbito de la computación y las redes se hace referencia a la misma como la ciencia y el arte de transformar mensajes para volverlos seguros e inmunes a ataques de terceros.

Puntualmente en la tecnología *Blockchain* se emplean los algoritmos conocidos como de clave asimétrica. Este tipo de criptografía presta diversos servicios a esta tecnología, entre los que se encuentran el no repudio, la integridad de los datos y la autenticación del origen de los mismos.

1.3. Componentes genéricos de la tecnología *Blockchain*

Existen diversas implementaciones de la tecnología *Blockchain*, sin embargo, más allá de las variaciones que se realicen en búsqueda de alguna optimización o adaptación a un caso de uso en particular, todas comparten un conjunto de componentes y arquitectura que la definen como tal. Es importante notar que *Blockchain* constituye sólo una capa dentro de la totalidad que conforma una aplicación.

Para el propósito del presente Trabajo Final de Graduación, y en estrecho vínculo con el objetivo específico de este capítulo, resulta importante partir del entendimiento sobre el uso y rol que juega la criptografía dentro de esta tecnología. Luego, corresponde analizar los elementos y estructuras de datos indispensables que componen este protocolo y garantizan su funcionamiento, siempre con la salvedad de mantener los conceptos alejados de cualquier implementación en particular.

1.3.1. Funciones hash criptográficas.

Como se mencionó en el apartado anterior la criptografía es fundamental en la tecnología *Blockchain*; el uso de la misma a través de funciones hash significa un componente notable para múltiples operaciones dentro de la misma.

En esencia, las funciones hash reciben una entrada de casi cualquier tipo y tamaño y devuelven una salida relativamente única y de tamaño fijo llamada comúnmente código hash, *digest* o simplemente hash.

La idea detrás de las funciones hash criptográficas es que, el código resultante sea una representación e identificador único e inequívoco de la entrada proporcionada a dicha

función, esta conducta es llamada propiedad determinista y es la responsable de que, ante un conjunto de datos dado, el código hash resultante sea siempre el mismo; siguiendo la lógica de esta propiedad, si la entrada original presenta el más mínimo cambio, el hash se vería modificado de manera radical.

Formalmente y citando a Alfred J, Menezes et al. en su libro Handbook of Applied Cryptography (1996), una función hash *h* mapea cadenas de bits de longitud finita arbitraria a cadenas de longitud fija, es decir de n bits.

Por otro lado, el motivo por el cual se habla de que las funciones hash presentan una salida relativamente única, está dado porque las posibles entradas son infinitas y las salidas un conjunto finito, es decir, para un dominio D y un rango R con h: D→ R y |D| > |R|, la función es de muchos a uno, lo que implica la existencia de colisiones (pares de entradas con salida idéntica), por lo tanto, es una situación inevitable aunque improbable (Menezes, van Oorschot, & Vanstone, 1996). Ante este escenario, las funciones hash criptográficas proporcionan mecanismos que mitigan al máximo las probabilidades de ocurrencia de colisiones.

Asimismo, otra condición fundamental es que la función hash criptográfica sea computacionalmente eficiente, es decir, el algoritmo matemático encargado de la obtención del valor hash debe resolverse con velocidad para no comprometer la performance de la aplicación que hace uso del mismo.

Por otra parte, existen 3 propiedades fundamentales que debe cumplir una función hash criptográfica para ser considerarse segura, tal como lo apunta el libro anteriormente

mencionado, Handbook of Applied Cryptography, (Menezes, van Oorschot, & Vanstone, 1996):

-Unidireccional o no reversible: esta propiedad hace referencia a la imposibilidad de arribar a los datos originales de entrada a partir del código hash asociado. Es decir, dado *digest y*, encontrar *x* donde la función hash(*x*) = *y*.

-Resistencia a la colisión débil: Las funciones hash criptográficas están diseñadas para que, dada una entrada específica, sea computacionalmente imposible encontrar una segunda entrada que produzca la misma salida. Esta condición se refleja mediante la imposibilidad de que dado *x*, encontrar *y*, tal que hash(*x*) = hash(*y*).

-Resistencia a la colisión fuerte: Es computacionalmente improbable encontrar dos entradas distintas a una función hash criptográfica que generen el mismo código hash de salida. Difiere de la anterior propiedad en que ambas entradas son de libre elección, esto es, se desconoce una referencia de entrada con su correspondiente salida. Dicho de otra manera, es imposible encontrar un *x* e *y* donde hash(x) = hash(y).

1.3.2. Criptografía Asimétrica.

La criptografía asimétrica tiene sus orígenes en 1976, en un artículo publicado por Whitfield Diffie y Martin Hellman, en el mismo sus autores proponen un sistema donde la distribución de claves criptográficas es posible a través de canales públicos sin comprometer la seguridad (Diffie & Hellman, 1976).

Para tal fin, este tipo de criptografía se vale de una clave pública, que puede ser distribuida a cualquiera sin riesgo, y otra privada, que sólo debe ser conocida por su

dueño; ambas están relacionadas matemáticamente entre sí, aunque a pesar de ello, no es posible determinar eficientemente la clave privada partiendo del conocimiento de la clave pública.

Su funcionamiento está basado en la encriptación a través de una de las llaves y la desencriptación mediante su par asociada. Estas "llaves", al igual que un código hash, son cadenas alfanuméricas de extensión determinada.

Como lo explican (Yaga, Mell, Roby, & Scarfone, 2018) la criptografía asimétrica dentro de la Cadena de Bloques provee confianza entre las partes que se desconocen. Esto lo hace consolidándose como un mecanismo que permite verificar la integridad y autenticidad de las transacciones a través de lo que se conoce como firmas digitales.

Entendiendo el principio que encierra este tipo de criptografía y en concordancia con lo expuesto en el informe del NIST (2018) se atribuyen principalmente 2 funcionalidades en la mayoría de las redes *Blockchain.* Por un lado y como se amplía a continuación, firmar transacciones digitalmente; y por el otro, la derivación de direcciones.

1.3.3. Firma digital.

Retomando el texto Handbook of Applied Cryptography (Menezes, van Oorschot, & Vanstone, 1996) se define clara y brevemente una firma digital como "una cadena de datos que asocia un mensaje (en formato digital) con alguna entidad originaria." En el caso de la tecnología *Blockchain* esto se logra combinando la llave privada del emisor,

representativa de la entidad originaria, y el hash de los datos a firmar (*digest* de la transacción) correspondiente al mensaje según la definición anterior. Claramente y en base a los conceptos de criptografía desarrollados previamente, para conseguir este propósito son necesarios tres algoritmos, uno para generar una llave privada al azar y su correspondiente llave publica; otro que se encargue de generar la firma como tal en base a la llave privada y el hash de los datos a firmar, y por último un algoritmo de verificación que permita autenticar la firma.

El conjunto de algoritmos de generación y verificación de firmas digitales es conocido como mecanismo o esquema de firma digital. La noción de un esquema de firma digital tiene sus orígenes junto a la criptografía asimétrica de la mano de Whitfield Diffie y Martin E. Hellman en 1976 en su paper ya citado New Directions in Cryptography.

En lo que respecta al funcionamiento de las firmas digitales es posible identificar 2 procesos que involucran los algoritmos de generación y verificación, estos son el de proceso de firmado digital, que consiste tanto en la generación de la firma, como en el formateo de los datos para ser firmados y, el proceso de verificación de firma digital, el cual tiene como objetivo la verificación de la firma y la recuperación del contenido firmado (Menezes, van Oorschot, & Vanstone, 1996).

En un ejemplo sencillo para entender la operatoria de una firma digital en una transacción podemos tomar a dos sujetos, sujeto A y sujeto B. Por su parte el sujeto A en su rol de remitente del mensaje o iniciador de la transacción ya cuenta con su par de llaves criptográficas, esto dentro del ecosistema *Blockchain* es realizado por un servicio

externo, siendo en el caso de las criptomonedas, con la creación una cartera digital. Los datos de la transacción provistos por A se cifran mediante un algoritmo que, tomando como ejemplo el estándar más empleado SHA-256, genera un *digest* de 64 caracteres. Este hash es combinado con la llave privada de A en lo que se conoce como el proceso de firmado digital. Posterior a este proceso se envían al receptor, B, los datos de la transacción, la firma digital (combinación de datos más la llave privada de A) y la llave pública de A. Una vez recibidos estos datos se inicia el segundo proceso, el de verificación de la firma digital; en esta instancia, el sistema por parte del sujeto B hace uso de la llave pública provista y perteneciente a A para descifrar la firma digital (sin vulnerar la clave privada de A) y recupera el hash de 64 caracteres correspondiente a los datos de la transacción cifrados por A en base a SHA-256. Como el sujeto B también recibió los datos de la transacción el sistema procede a cifrarlos en SHA-256 para obtener el hash correspondiente, que como se vio en el apartado de funciones hash criptográficas, debería ser el mismo que el obtenido al verificar la firma digital, cumpliendo la propiedad determinista de estas funciones; en caso de diferir, esto indicaría que los datos fueron alterados o que la clave pública provista por A no corresponde con su clave privada, volviendo automáticamente inválida la transacción.

Existen diversos esquemas de firma digital, sin embargo, el que se emplea mayormente en la tecnología *Blockchain*, es el estándar para firmas digitales del NIST

llamado *Elliptic Curve Digital Signature Algorithm* (ECDSA)11 que toma la matemática detrás de los campos finitos y las curvas elípticas para generar las llaves.

Por otro lado, Menezes et al. (1996) indica que los esquemas seguros de firmas digitales deberían proveer las propiedades de autenticidad, integridad de datos y no repudio.

1.3.4. Transacción.

Una transacción es una unidad de datos fundamental para la Cadena de Bloques y refleja las operaciones realizadas sobre la misma.

Las transacciones agrupan datos firmados digitalmente y se transmiten por la red, a medida que son validadas por los nodos, se juntan y ordenan para formar los bloques que conforman la cadena.

La información que almacenan estas transacciones difiere en cada Cadena de Bloques y su utilización, por ejemplo, en la *Blockchain* de Bitcoin las transacciones representan la transferencia de valor (criptomonedas) de un usuario a otro, mientras que en una *Blockchain* avocada a otros casos de uso podrían significar el registro de movimientos sobre activos físicos o digitales o simplemente la publicación de datos de forma permanente y abierta.

---

[11] National Institute of Standards and Technology (NIST), Digital Signature Standard, Federal Information Processing Standards (FIPS) Publication 186-4, July 2013. https://doi.org/10.6028/NIST.FIPS.186-4

Mas allá del tipo de información existen una serie de datos indispensables al momento de transmitir una transacción, en su mayoría fueron vistos en el apartado de firma digital y están compuestos por: los datos de la transacción, que dependiendo el uso y la implementación particular de una *Blockchain*, puede incluir salidas y entradas (ambos conceptos propios de la transacción de criptomonedas), una marca de tiempo, la clave pública, la firma digital y direcciones; estas últimas son derivaciones de las claves públicas y se usan como puntos finales, desde y para, en una transacción.

1.3.5. Árbol de Merkle.

Cuando se abordó la historia y los principios que llevaron al desarrollo de *Blockchain* como la conocemos, se mencionó el árbol de Merkle como un avance fundamental en el campo de esta tecnología. Conociendo el concepto de hash y transacciones, es posible dar el paso y adentrarse en el arbol de Merkle y su rol dentro de una *Blockchain.*

Las transacciones, como se explicó con anterioridad, se encuentran cifradas por hashes; el conjunto de *digest*s que las representan, hace uso de una estructura de datos en forma de árbol, conocido como árbol hash o árbol de Merkle para resumirse y ordenarse de forma estricta a medida que crecen en cantidad, brindando un método seguro, veloz y ligero para verificar los datos contenidos en las mismas. Esta estructura de datos, en analogía a un árbol (visto al revés), simula un orden de jerarquía descendente, partiendo de un único valor llamado raíz del cual se disocian y desprenden valores hoja, todos ellos correspondientes a los códigos hash representativos de las transacciones. El propósito

fundamental de esta estructura radica en la creación del código raíz, el cual es un resúmen de todos los valores hojas y se consigue agrupando de manera recursiva pares de hash correspondientes a transacciones hasta que solo queda un código hash (raíz de Merkle).

1.3.6. Consenso.

El consenso es un concepto tomado de los sistemas distribuidos y por lo tanto aplicado en la tecnología *Blockchain*. Consiste en el protocolo mediante el cual los nodos de la red, que se desconocen y por consiguiente no confían entre sí, llegan a un acuerdo sobre el valor final de los datos almacenados y el estado de la Cadena de Bloques. Si bien lograr el acuerdo entre dos nodos no resulta un desafío, lo notamos en la conocida arquitectura cliente servidor, cuando se trata de un sistema distribuido donde múltiples nodos intervienen en la definición de un único valor, el objetivo se vuelve difícil de conseguir (Bashir, 2018). Es por este rol clave que el consenso se puede considerar el cerebro de la tecnología *Blockchain*, sin la existencia de este componente estaríamos en presencia de una base de datos inmutable mediante la protección criptográfica de los registros allí almacenados (algo factible de conseguir mediante otras tecnologías).

Dentro de la tecnología *Blockchain* el protocolo de consenso es el responsable de garantizar la confianza entre los nodos que se desconocen; al crearse la cadena se inicia con un bloque génesis el cual esta preconfigurado y define el estado inicial del sistema, los siguientes bloques que se agregan lo hacen acorde al protocolo de consenso definido para dicha implementación. En este sentido, la combinación de un estado inicial sumado a la capacidad, ya analizada, de los nodos de verificar cada bloque publicado desde

entonces, permite a los usuarios, acordar independientemente el estado actual de la *Blockchain* (Yaga, Mell, Roby, & Scarfone, 2018).

Yaga et al. (2018) propone una serie de propiedades que resultan supuestos ante la utilización de un protocolo de consenso en un sistema *Blockchain,* entre ellos menciona la existencia de un estado inicial en el que los nodos participantes estén de acuerdo; los usuarios, por su parte, comprenden y aceptan el modelo de consenso que empleara el sistema para agregar nuevos bloques; debe existir la vinculación entre los bloques, esto se consigue mediante un puntero hacia el hash del encabezado del bloque anterior, siendo la excepción el bloque génesis; gracias al encadenamiento y el protocolo, los usuarios pueden verificar cada bloque de manera independiente.

Por otro lado, bajo un criterio similar al de Yagal et al., Bashir (2018) propone, abstrayéndose de su uso en la Cadena de Bloques, requerimientos que un mecanismo de consenso debe cumplir para conseguir el resultado esperado dentro de un sistema distribuido:

- Acuerdo: El consenso logrado para decidir un estado o valor debe depender del acuerdo entre los nodos honestos;

- Finalización: Como cualquier algoritmo el mecanismo de consenso debe contar con una finalización. Todos los nodos participantes deben dar por finalizado el proceso y eventualmente llegar a una decisión.

-Validez: El valor acordado por todos los nodos honestos, posterior al proceso de conceso, debe ser el mismo que el propuesto inicialmente por al menos uno de los nodos honestos en la red.

-Tolerancia a fallos: El algoritmo de consenso debe ser tolerante a fallos, es decir, debería ejecutarse ante la presencia de nodos bizantinos, ya sea defectuosos o mal intencionados.

- Integridad: Cada nodo participante puede decidir solo una vez acerca de un estado o valor durante el ciclo del protocolo de consenso.

1.3.7. Bloque.

Según la empresa española Bit2Me especializada en *Bitcoin* y *Blockchain*, un bloque dentro de la tecnología *Blockchain* es un concepto pensado para optimizar el proceso de validación de transacciones (Bit2Me).

En definitiva, un bloque es una estructura de datos de tamaño variable conformada por un conjunto de transacciones y metadatos. Tanto los datos (transacciones) como metadatos almacenados en un bloque difieren acorde a las implementaciones de esta tecnología, sin embargo, al analizar los bloques correspondientes a las *Blockchain*s de Bitcoin[12] y Ethereum[13], y en acuerdo con lo expuesto por Yaga et al. (2018), se puede arribar a la generalización de un bloque conformado por dos partes, una cabecera con metadatos y los datos o transacciones propiamente dichos. La cabecera, almacena al menos los siguientes datos:

[12] https://blockexplorer.com/blocks

[13] https://etherscan.io/blocks

- El número de bloque, éste es un número correlativo al anterior que aumenta a medida que los bloques van sumándose a la cadena, es también conocido como altura por esta misma razón.

- El hash correspondiente a la cabecera del bloque anterior, este hash es el que mantiene la integridad de la cadena y funciona como puntero para enlazar los bloques.

- Una representación de los datos almacenado, que como ya se explicó, para este fin se emplea la raíz de un árbol de Merkle en función de sintetizar las transacciones contenidas en el bloque y agilizar su proceso de verificación y validación.

- Una marca de tiempo que indica cuando el bloque fue añadido a la cadena.

- El tamaño del bloque, por lo general expresado en bytes.

- Por último, un valor conocido como n*once,* a diferencia de los anteriores, no es un dato imprescindible, aunque presente en la mayoría de las implementaciones por su utilidad en el proceso de consenso, es un numero generado y usado por única vez en una suerte de rompecabezas de hashes para la publicación de un nuevo bloque en la cadena.

Es importante aclarar que los datos almacenados en esta cabecera son sometidos a una función hash criptográfica que genera el *digest* correspondiente al bloque, que luego es usado como puntero durante el encadenamiento.

Por otra parte, el cuerpo del bloque está constituido por el conjunto de transacciones y eventos registrados dentro del mismo, pudiendo en algunos casos presentar otros datos adicionales.

1.4. Encadenando los bloques

Teniendo conocimiento sobre los componentes genéricos de una *Blockchain*, es momento de introducirse en como ellos se relacionan para que esta tecnología funcione, validando transacciones, creando bloques y agregándolos a la cadena.

Se presenta a continuación un esquema general de creación y encadenado de bloques teniendo en cuenta la relación que guardan con las transacciones contenidas en ellos. Para tal fin resulta de suma utilidad una interpretación y traducción libre del sumario propuesto por (Bashir, 2018).

Es oportuno aclarar la omisión de ciertos detalles y conceptos que fueron introducidos con anterioridad; en caso de no comprender la esencia de alguno de los términos a continuación, se recomienda regresar la lectura al apartado correspondiente. Por otro lado, advertir que mucho de los procedimientos llevados a cabo por esta tecnología dependen exclusivamente del mecanismo de consenso empleado, por lo tanto, se procura arriba a un entendimiento de su funcionamiento sin involucrar especificidades propias de sus diversas implementaciones.

1. Un nodo inicia una transacción creándola y firmándola digitalmente.
2. La transacción es propagada al resto de los nodos de la red p2p a través de un protocolo Gossip, donde es validada en base a criterios predefinidos y retrasmitida a sus nodos pares; el conjunto de transacciones que son escuchadas por los nodos se registra en un pool de transacciones, el cual puede diferir entre nodos a cusa de los diferentes tiempos de escucha.

3. Una vez seleccionado el nodo publicador a través del mecanismo de consenso (concepto que es analizado posteriormente), el pool de transacciones se constituye junto a datos adicionales como un bloque, el cual ahora es transmitido a toda la red. En esta instancia se considera que la transacción fue confirmada.

4. El nuevo bloque es validado y agregado a la cadena por el resto de los nodos de la red. El bloque se enlaza a la cadena haciendo uso de la criptografía y el hash correspondiente al encabezado del bloque anterior.

Es importante remarcar sobre el último paso que, si existiera un cambio en un bloque previamente publicado, tendría un hash de encabezado diferente; esto, a su vez, causaría que todos los bloques subsiguientes también tengan hashes diferentes, ya que incluyen el hash del bloque anterior como parte de los datos de entrada para producir su propio hash. Gracias a este mecanismo es posible detectar y rechazar fácilmente los bloques alterados proporcionando un sistema de verificación constante de las transacciones ya almacenadas y por almacenarse.

En la siguiente ilustración recuperada de un curso provisto por Linux Fundation en la plataforma edX llamado *Blockchain* for Business - An Introduction to Hyperledger Technologies, se observa el encadenamiento de bloques y el árbol de Merkle que relaciona las transacciones en una raíz utilizada para garantizar la inmutabilidad.

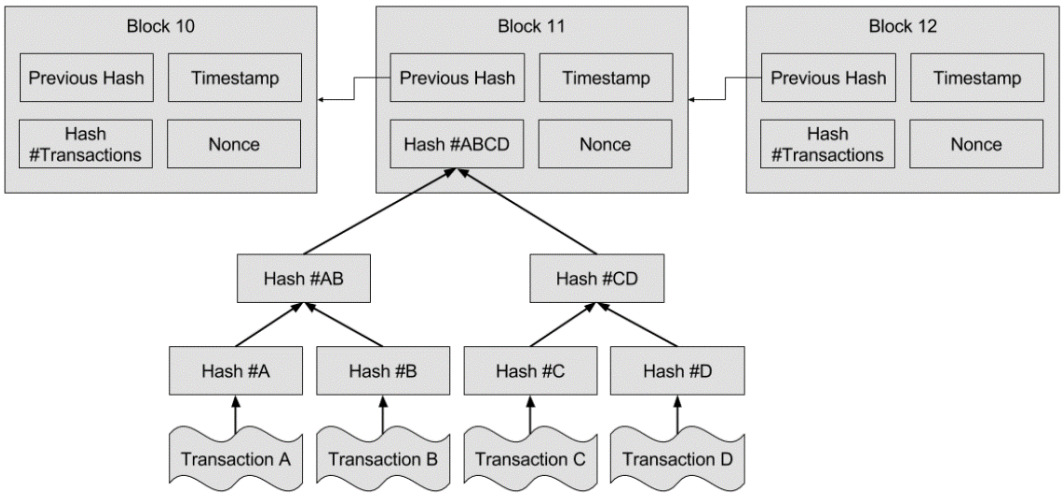

Figura 4. Estructura de una Cadena de Bloques.
Fuente: edx.org, Curso en Hyperledger

1.5. Tipos de *Blockchain*

Con una idea clara de lo que implica la tecnología *Blockchain*, podemos categorizarlas en base al modelo de permisos que adoptan, es decir sobre quiénes son los encargados de mantener y hacer uso de la red. Es relevante esta distinción debido a que el comportamiento de la tecnología diferirá sustancialmente acorde al modelo adoptado y tendrá una clara repercusión en el caso de uso al cual se busque aplicar *Blockchain*. Si bien en relación a este aspecto existen disparidades entre las categorías consideradas por los distintos autores consultados, se perciben dos criterios empleados para su diferenciación, por un lado, en base al acceso a la red y los datos almacenados, y por el otro basándose en las capacidades y responsabilidades de un nodo dentro de la red. Respecto al primer criterio se distinguen tres tipos acorde a la autenticación de los nodos en la red, las Cadenas de Bloques públicas o abiertas, privadas o cerradas y federadas o

de consorcio; en cuanto a las funcionalidades de validación que tiene autorizado un nodo sobre la red nos encontramos con *Blockchains* permisionadas y sin permisos.

Además de esta taxonomía, amerita mencionar, en este capítulo, un concepto introducido recientemente por importantes firmas entre las cuales se puede nombrar Amazon, Microsoft e IBM, conocido como *Blockchain* como servicio (*Blockchain* as Service - BaaS).

Blockchain pública.

Como su nombre lo indica, este tipo de Cadenas de Bloques no pertenecen a alguien en concreto, están abiertas a cualquier individuo. En ella, quienes deciden formar parte de la red, cuentan con los mismos derechos y deberes a la hora de proponer y validar transacciones que el resto de los nodos que la conforman. Alexander Preukschat en su libro *Blockchain* la Revolución Industrial de Internet (2017), apunta que en este tipo *Blockchain* cualquier usuario sin ser parte de la red puede consultar las transacciones allí registradas, por otro lado, si un usuario desea convertirse en un nodo validador, tiene la libertad de hacerlo y participar del protocolo común establecido para dicha Cadena de Bloques. Además, destaca la igualdad de condiciones de los nodos en la red mediante la inexistencia de un usuario con mayor autoridad que el resto sobre las decisiones tomadas en la misma. Por otro lado debido a su carácter público los usuarios tienden a desconocerse y actúan de manera pseudoanónima, ya que, si bien no pueden ser identificados personalmente, es posible rastrearlos a través de las direcciones con las que operan.

Está claro que este tipo *Blockchain* recupera la esencia y el propósito de su creación de la mano de Bitcoin; es también evidente el rol fundamental de los algoritmos de Consenso para generar confianza entre las partes y el correcto funcionamiento del sistema.

Blockchain privada.

Una *Blockchain* privada pertenece a una organización o grupo de individuos donde sólo son partícipes de la red aquellos usuarios invitados y autorizados por quien es responsable de la cadena; por esta misma razón es que lo datos allí almacenados, por lo general, no son de libre acceso al público general.

Una entidad se encarga de mantener la cadena, aceptar nuevos miembros y velar por su correcto funcionamiento. Aquí el anonimato desaparece ya que los nodos, al requerir su aceptación e invitación previa para formar parte de la red, son conocidos.

En este tipo *Blockchain* se desdibuja el concepto de descentralización, ya que una organización es la única encargada de tomar decisiones para la validación de transacciones y generación de los bloques, respondiendo siempre a sus intereses propios.

Blockchain federadas o de consorcio.

Las *Blockchains* federadas son híbridos que buscan reunir los mejores aspectos de las *Blockchains* públicas y privadas; utópicamente pretenden valerse de acceso controlado y libertad al mismo tiempo.

Las *Blockchains* hibridas dan respuesta a la necesidad de registros compartidos y distribuidos por entidades vinculadas que presentan diferentes intereses. Bajo este enfoque se percibe el carácter privado de la red, siendo responsables de su mantenimiento y administración un número determinado de organizaciones, las cuales utilizan nodos pre seleccionados para su operatoria.

Por lo general este tipo de *Blockchain* presentan una interfaz de acceso al usuario medio que limita el carácter público y abierto de la totalidad de los datos.

A pesar de ser muy similares a las Cadenas de Bloques privadas, las federadas eliminan la influencia de una organización específica sobre la red, buscando la sinergia entre varias organizaciones que se autorregulan entre sí.

Blockchain sin permisos.

Si bien se plantea como una categoría diferente a las 3 anteriores, tiene una relación directa y suele emplearse de manera indistinta para hacer referencia a las Cadenas de Bloques públicas. En esencia una *Blockcahin* de este tipo no requiere del consentimiento de una autoridad o ente particular para formar parte e intervenir en el funcionamiento y mantenimiento de ella. Cualquier usuario puede realizar operaciones de lectura, escritura y validación de transacciones dentro de la Cadena de Bloques.

Cadenas de Bloques permisionadas.

Situación similar a la explicada en las *Blockchains* sin permisos. Se trata de una cuestión de enfoque, en este caso, la categoría es mayormente ligada a las redes privadas; existe una autoridad que atribuye diferentes permisos y funcionalidades dentro de la red a

los nodos participantes; es así como algunos de ellos están limitados a la lectura, otros a la publicación de transacciones o incluso autorizados a funcionar como nodos validadores. En otras palabras, las responsabilidades y participación de los nodos para con la red es dependiente de los permisos que se les atribuya (Ver tabla1).

A continuación, se presenta una tabla de elaboración propia que simplifica la mirada y relación entre las dimensiones anteriormente descriptas para clasificar las categorías de Cadenas de Bloques. Si bien se excluyen las de tipo federadas o de consorcio en los ejes, como se mencionó con anterioridad, estas en realidad son producto de combinar características de otras tipologías puras, por lo cual lo apropiado es ubicarlas en la intersección de propiedades producto de las dos taxonomías propuestas.

Tabla 1. *Tipologías Blockchain.*

Acceso a las transacciones	***Acceso al envío y validación de transacciones***	
	Con Permisos	Sin Permisos
Pública	Todos pueden leer los registros de la cadena peros solo nodos autorizados pueden enviar transacciones y otros validarlas. *Estas características se asocian generalmente a la tipología Blockchain federada o de consorcio.	Todos los nodos que lo precisen pueden leer, enviar y validar transacciones.
Privada	Sólo los nodos autorizados pueden cumplir con el rol de leer, enviar y/o validar transacciones.	No aplica.

Blockchain como servicio.

Este tipo *Blockchain* es ofrecido por empresas como alternativas basadas en la nube que permiten a los clientes crear, usar y alojar sus propias soluciones *Blockchain* delegando a la empresa proveedora del servicio las tareas relacionadas a mantener la infraestructura ágil y operativa, simplificando el proceso de creación y administración de la red. Por lo general este tipo de servicios habilita la creación de *Blockchains* de tipo privadas o permisionadas.

1.6. Aplicaciones distribuidas y *Blockchain*.

Esta sección se enfoca en como la tecnología *Blockchain* puede ser utilizada para la construcción de aplicaciones distribuidas sobre la misma. Para ello, se parte de lo conocido como generaciones o etapas evolutivas de *Blockchain,* para arribar posteriormente a cuestiones más técnicas como la diferencia entre una app tradicional y una de tipo distribuida.

1.6.1. Evolución de la tecnología *Blockchain*.

Swan (2015) diferencia a lo largo de su libro *Blockchain*- Blueprint for a New Economy, 3 categorías o generaciones en las que se pueden agrupar las diferentes Cadenas de Bloques acorde al soporte que brindan para el desarrollo de diferentes aplicaciones. Esta clasificación fue gestándose en el tiempo con los avances y la evolución de las capacidades y prestaciones introducidas en la *Blockchain*; es entonces que se puede hablar de *Blockchain* 1.0, 2.0, y 3.0.

Blockchain 1.0 hace referencia a la primera implementación de la mano de Bitcoin, por lo tanto, esta generación se vincula directamente con las criptomonedas y las aplicaciones que posibilitan el registro y la transferencia directa de las mismas.

Blockchain 2.0, por su parte, son los contratos inteligentes, su aparición vino de la mano de Ethereum, una Cadena de Bloques que incorporo a su núcleo la tecnología de *Smart Contracts*; Etherum creó un intérprete de lenguaje de programación más extenso (Turing completo[14]) que el presente en la implementación de Bitcoin, esto permitió agregar lógica mucho más compleja dentro de la *Blockchain.* Este nuevo componente posibilito la ampliación del abanico de aplicaciones factibles de ser desarrolladas, permitiendo no sólo transacciones directas de criptomonedas sino incluir el tratamiento de otros bienes conocidos como *assets*.

Por último, se encuentra la generación 3.0, esta generación es la que está actualmente en auge y persigue acabar con los problemas de escalabilidad, interoperabilidad y performance que acarrean las generaciones anteriores. Si bien, con Ethereum se empezó a percibir la idea de *Blockchain* como servicio, la generación 3.0 se sostiene firmemente sobre esta idea, así como en la construcción de aplicaciones distribuidas (DApps). Swan (2015) por su parte, atribuye esta generación esencialmente a los casos de uso que exceden a los financieros, de criptomonedas, económicos y de mercados; sin embargo, a raíz de los avances dados en el campo de esta tecnología, que al momento de la edición de su libro no estaban presentes, resulta más apropiado agrupar en

[14] La expresión “Turing Completo” hace alusión a los sistemas lógicos o lenguajes de programación capaces de realizar cualquier cálculo computacional con los recursos adecuados.

esta categoría a aquellos proyectos que exhiben un cambio sustancial en la infraestructura tecnológica. Este cambio puede estar dado ya sea por nuevos protocolos de consenso y comunicación o por arquitecturas completamente innovadoras que permiten acabar con las limitaciones mencionadas con anterioridad. Estas mejoras, y en acuerdo con Swan, son posibilitadoras de nuevos casos de uso en un plano mayormente de carácter empresarial u organizacional. Entre estas nuevas infraestructuras se encuentran los grafos acíclicos dirigido (DAG), un tipo de Registro Distribuido que está ganando muchos adeptos, como una mejor alternativa a *Blockchain*.

1.6.2. *Smart Contracts*.

El término contrato inteligente data de 1994, definido por Nick Szabo como un protocolo de transacción computarizado que ejecuta los términos de un contrato. Los objetivos generales en el diseño de un contrato inteligente son: satisfacer condiciones contractuales comunes (como condiciones de pago, derechos de retención, confidencialidad e incluso cumplimiento), minimizar las excepciones maliciosas y accidentales, y minimizar la necesidad de intermediarios de confianza (Szabo, 1994).

Como señala el NIST, los contratos inteligentes extienden y aprovechan la tecnología *Blockchain*. Estos están formados por una colección de código y datos (a veces denominados funciones y estados) que se implementan mediante transacciones firmadas criptográficamente en la red *Blockchain*. El contrato inteligente es almacenado y ejecutado por nodos dentro de la red; es fundamental que sea cual fuera el nodo que

ejecute el contrato inteligente deriven los mismos resultados, esto evidencia la propiedad determinista de estos protocolos.

Los usuarios de la red *Blockchain* pueden crear transacciones que envían datos a funciones públicas ofrecidas por un contrato inteligente. El contrato inteligente ejecuta el método apropiado con los datos proporcionados por el usuario para realizar un servicio. El código, que se encuentra en la Cadena de Bloques, es transparente a los usuarios e inmutable, por lo tanto, puede utilizarse (entre otros fines) como un tercero de confianza. Un contrato inteligente puede realizar cálculos, almacenar información, exponer propiedades y, si corresponde, enviar automáticamente fondos a otras cuentas. No necesariamente tiene que realizar una función financiera (Yaga, Mell, Roby, & Scarfone, 2018).

1.6.3. Aplicaciones distribuidas.

Las aplicaciones distribuidas son esencialmente aplicaciones formadas por diversos componentes que se ejecutan en entornos separados e interactúan gracias a la conexión en red existente entre ellos. Es así que, arquitecturas bien conocidas y ampliamente adoptadas en el desarrollo de software moderno como la arquitectura cliente-servidor o cliente-middleware-servidor, cumplen con la anterior premisa y pueden ser consideradas dentro de este grupo. Sin embargo, volviendo al foco puesto en la tecnología *Blockchain*, existen sutiles pero importantes diferencias en cuanto a la concepción de las DApps.

En el entorno de la Cadena de Bloques las aplicaciones distribuidas entran en escena, como se mencionó anteriormente, con la aparición de Ethereum, allá por el año 2014. En este sentido Ethereum dotó a la tecnología de instrumentos para lograr desplegar aplicaciones que se ejecuten sobre ella; los avances más importantes que introdujo en escena fueron el lenguaje de programación Solidity, un lenguaje especialmente orientado al desarrollo de contratos inteligentes o *Smart Contracts*, y la Máquina Virtual de Ethereum (EVM). En un alto nivel de abstracción, el EVM es el componente de Ethereum que se encarga de gestionar el despliegue y la ejecución de contratos inteligentes a través de nodos interconectados mediante una red *peer to peer*.

Es notorio, a raíz de lo expuesto con anterioridad, que la diferencia fundamental entre una DApp sobre una arquitectura cliente-servidor y una basada en *Blockchain* se encuentra fundamentalmente en conseguir un entorno de ejecución y almacenamiento completamente descentralizados, diluyendo la idea de un servidor único encargado de proveer recursos o servicios.

Ahondando en las diferencias y atendiendo al diseño, a un alto nivel de abstracción, una aplicación web consta de 3 capas acordes a una separación de intereses y componentes: una capa de acceso a datos conocida como back-end, encargada de procesar la lógica de negocio; una capa de presentación, a la que suele referirse como front-end y se encarga de la interacción del usuario con la aplicación; y por último una capa de persistencia de datos donde se encapsulan los comportamientos necesarios para mantener los objetos de manera permanente o casi permanente en una base de datos. En el caso de una spp basada en *Blockchain*, las capas que se mencionaron con anterioridad

sufren ligeras modificaciones; existe una unión de intereses acaparada por esta tecnología donde tanto el back-end como la persistencia se agrupan bajo las prestaciones de la Cadena de Bloques*;* la lógica de negocios recae ahora en gran parte en la función de los contratos inteligentes, mientras que el sistema de almacenamiento o base de datos, es sustituido por la *Blockchain* en su acepción como estructura de datos.

En lo que respecta la aplicación cliente, es posible emplear los mismos lenguajes para el desarrollo que en una aplicación web tradicional, como por ejemplo el stack MEAN js (MongoDB, ExpressJS, AngularJS, Node.js), empero, un cambio importante se manifiesta en la manera en que se integra esta capa con la correspondiente a los componentes *Blockahin*, es decir, la manera en que se comunican la aplicación cliente y los contratos inteligentes. Esta particularidad en la comunicación entre capas, es producto del carácter descentralizado de la tecnología *Blockchain* y el funcionamiento de los Contratos Inteligentes.

Bajo el dominio de esta tecnología cualquier operación exceptuando la lectura, es tratada como una transacción, incluyendo el despliegue, la interacción y ejecución de los Contratos Inteligentes.

Para lograr su cometido, lo primero que debe hacer la aplicación cliente es, comunicarse con un nodo activo perteneciente a la red de la Cadena de Bloques, para ello hace uso de un API (Application Programming Interface) especialmente desarrollada para para tal fin, siendo la más conocida y a modo de ejemplo, la API desarrollada en Javascript para la plataforma Ethereum, Web3. Además, se vale de la especificación conocida como JSON-RPC para llamadas a procedimientos remotos. Cuando el usuario

realiza las transacciones a través de la API, el nodo al que se conecta las incluye en un pool de transacciones a ser verificadas y validadas. Una vez minado ese bloque se propaga por toda la *Blockchain* como se explicó en el correspondiente apartado.

Es importante destacar que, la API Web3 o cualquier semejante, es necesaria durante el desarrollo del contrato inteligente para lograr su despliegue y hospedaje en la Cadena de Bloques para posteriormente ser ejecutado en ella, volviéndose esta interfaz un componente imprescindible como punto de acceso a dicha tecnología.

En la figura 5 se puede observar el proceso descripto anteriormente, donde una aplicación cliente, escrita en cualquier lenguaje bien conocido como JavaScript, Java o Python, se comunica a través de la API Web3 y el estándar para llamada a procedimientos remotos, con la red *Blockchain* de Ethereum.

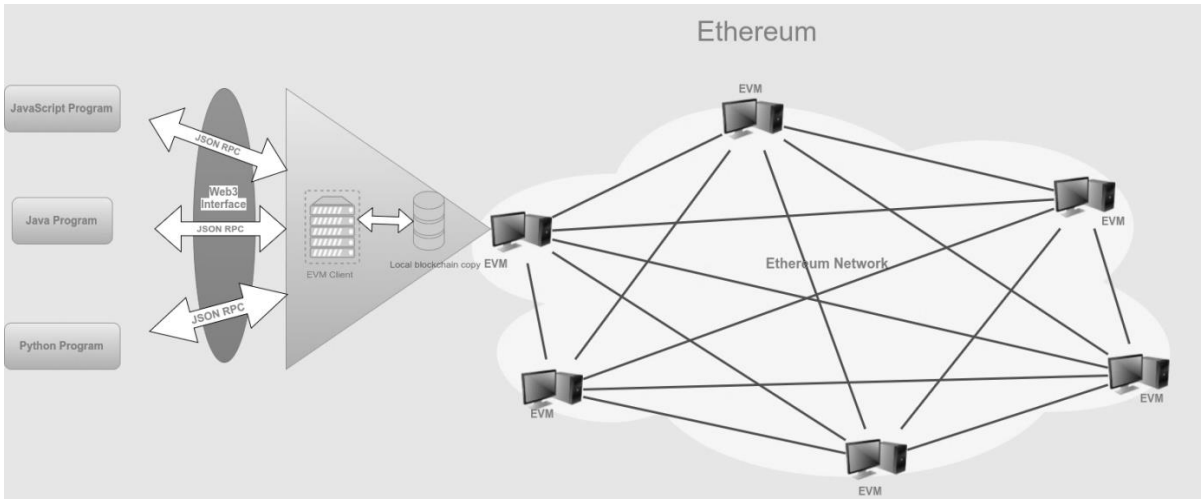

Figura 5. Conexión de una aplicación cliente con la *Blockchain* Ethereum.
Imagen recuperada de http://www.DAppuniversity.com/articles/web3-js-intro

1.7. Observaciones finales sobre la tecnología *Blockchain*.

La tecnología *Blockchain*, como se pudo ver a lo largo del capítulo, combina conceptos, técnicas y herramientas conocidas de las ciencias de la computación para dar origen a propiedades emergentes únicas. Son estas propiedades las que dotan de beneficios a los sistemas que incorporan esta tecnología como componente esencial; sin embargo, las mismas son responsables de que exista la necesidad latente de una redefinición de procesos de ingeniería en software existentes para conseguir la adaptación a dichas particularidades.

Algunos puntos clave a tener en cuenta sobre la tecnología *Blockchain* a la hora de pensar en la falta de ajuste de los procesos de desarrollo existentes son:

- La tecnología *Blockchain* reporta beneficios solo en determinados casos de uso. Por ese motivo el tratamiento de los requerimientos del sistema debe enfocarse primeramente en analizar si la tecnología *Blockchain* verdaderamente agrega valor a la solución.

- El comportamiento de la *Blockchain* varía según la tipología y la plataforma seleccionada para llevar adelante la implementación. Es imperativo planificar y evaluar estos aspectos antes de embarcarse en el desarrollo del proyecto.

- No es posible realizar un CRUD como en una base de datos tradicional, ya que el comportamiento de la *Blockchain* sólo permite agregar registros. Sumado a esto, la propiedad de inmutabilidad de la Cadena de Bloques, condiciona el diseño de los datos allí almacenados, así como la necesidad de que la aplicación cliente mitigue el posible registro de datos erróneos.

- Los *Smart Contracts* al ser tratados como transacciones, cumplen con la anterior observación en cuanto a la inmutabilidad, por lo tanto, es necesario asegurar su correcto funcionamiento previo al despliegue productivo en la *Blockchain* principal.

- Al centrarse la lógica de las aplicaciones *Blockchain* en los *Smart Contracts*, estos deben recibir la mayor atención posible en su diseño y desarrollo. UML no contempla estas entidades ni su tratamiento por lo tanto el modelado se vuelve un desafío.

- Las actividades de pruebas tienen que ser una prioridad frente a un desarrollo que posterior a su despliegue no puede ser modificado.

- En el caso de precisar una *Blockchain* 100% personalizada son muchos y complejos los componentes a ser considerados para el desarrollo (criptografía, redes peer to peer, sistemas distribuidos, tolerancia a fallos, mecanismos de consenso). Por ese motivo resulta mejor alternativa utilizar como base para un desarrollo de la infraestructura *Blockchain*, plataformas ya consolidadas donde existe soporte y cierta estandarización en las prácticas de desarrollo como por ejemplo Ethereum.

Capítulo 2.

Prácticas de desarrollo *Blockchain* en Argentina

Este capítulo tiene como objetivo lograr un acercamiento y evaluar la realidad de las prácticas de desarrollo *Blockchain* llevadas adelante por la comunidad de profesionales del Software en Argentina. Si bien se cuenta con referencias sobre resultados de investigaciones en el mismo campo, como el trabajo realizado por Chakraborty, Shahriyar, Iqbal, & Bosu (2018) denominado Understanding the Software Development Practices of *Blockchain* Projects: A Survey; es oportuno, como factor que sustenta y valoriza el ideal del presente Trabajo Final de Graduación de investigación, hacer hincapié en el ecosistema que rodea esta tecnología en nuestro país.

2.1. Elaboración del instrumento.

Tomando como punto de partida el proceso especificado en el marco metodológico, se enuncian a continuación una serie de preguntas de investigación (PI) que representan el objetivo puntual de este capítulo; estas preguntas son las bases sobre las cuales se estructura y desarrolla el contenido del instrumento empleado para la recolección de datos.

PI 1. ¿Cuál es el perfil académico de los desarrolladores *Blockchain*?

PI 2. ¿Bajo qué proceso de desarrollo de software trabajan?

PI 3. ¿Qué prácticas se emplean en la fase de requerimientos?

PI 4. ¿Qué prácticas se emplean en la fase de diseño?

PI 5. ¿Qué prácticas se emplean en la fase de desarrollo?

PI 6. ¿Qué prácticas se emplean en la fase de pruebas?

PI 7. ¿Qué prácticas se emplean en la fase de mantenimiento?

PI 8. ¿Qué prácticas de gestión de configuración se emplean?

PI 9. ¿Qué prácticas para la gestión del proyecto son empleadas?

PI 10. ¿Cómo realizan el aseguramiento de la calidad a lo largo del proceso?

Cada una de las anteriores preguntas se desglosan dentro de la encuesta para facilitar su posterior análisis.

A continuación, se presenta una grilla con las preguntas de la encuesta[15] asociadas a su área de interés y PI. El tipo de respuesta denota el enfoque de las preguntas; es propicio aclarar la incorporación de un espacio que brinda la posibilidad de ingresar un comentario al final de cada sección de la encuesta. Por otro lado, los ítems de selección, a no ser que sean binarios (Si, No) incluyen siempre un campo "otra" que puede ser completado en caso de no encontrarse la alternativa precisada por el encuestado. En cuanto a las escalas, se discretizan en números del 0 a 5 los llamados ítems de Likert; la escala toma los siguientes valores, en el caso de medir frecuencia: 0=Nunca, 1=Raras veces, 2= Ocasionalmente, 3=Frecuentemente, 4=Muy frecuentemente; en el caso de medir importancia: 0=Sin importancia, 1=Poca importancia, 2= Moderadamente importante, 3=Importante, 4=Prioridad

[15] La encuesta con la totalidad de opciones en su formato original de distribución, y las respuestas recibidas, pueden ser visualizadas en el siguiente enlace: https://bit.ly/2XKjVsV

PI	Aspecto	Pregunta de encuesta	Tipo de respuesta			
			Selección única	Selección múltiple	Escala	Texto libre
1	Demográfico	1. ¿De qué provincia argentina eres?	x			
		2. ¿Eres profesional de las ciencias de la computación?	x			
		3. ¿Cuánto tiempo de experiencia tienes en el desarrollo de software?	x			
		4. ¿Cuánto tiempo llevas involucrado en el desarrollo de aplicaciones *Blockchain*?	x			
		5. ¿Desarrollas en *Blockchain* de manera profesional o como hobby?	x			
		6. ¿Con que tipo de formación relativa a *Blockchain* cuentas?	x			
		7. ¿Formaste parte de un proyecto basado en *Blockchain* desde sus inicios?	x			
		8. ¿Sobre qué plataforma *Blockchain* trabajas?		x		
		9. ¿El proyecto en el que trabajas tiene origen en Argentina?	x			
2	Proceso general de desarrollo	10. ¿Cuál de los siguientes procesos de desarrollo de software es utilizado?	x			
		11. ¿A qué fases del ciclo de vida del software dedican mayor atención?		x		
3	Fase de Requisitos	12. ¿Con que frecuencia se realizan prácticas relativas a los requisitos en su proyecto?			x	
		13. ¿Cuáles de las siguientes prácticas se aplican en relación a los requisitos?		x		
		14. ¿A través de que herramientas se especifican y analizan los requisitos?		x		
		15. Observaciones acerca de los requisitos en entornos *Blockchain.*				x
4	Fase de análisis y diseño	16. ¿Con que frecuencia se realizan prácticas relativas al diseño en su proyecto?			x	
		17. ¿Utilizan el lenguaje unificado de modelado (UML) u otra notación para el diseño del sistema?	x			
		18. ¿Cuáles de los siguientes modelos del sistema se documentan?		x		
		19. ¿Qué importancia atribuyen a cada uno de los factores de calidad del software al momento de diseñar el sistema?			x	

		20. Observaciones acerca del diseño en entornos *Blockchain.*				x
5	Fase de desarrollo	21. ¿Se definió algún estándar de codificación en el proyecto?	x			
		22. ¿Cuáles de las siguientes prácticas se aplican durante el desarrollo?		x		
		23. Observaciones acerca de la programación en entornos *Blockchain*				x
6	Fase de Pruebas	24. ¿Existe un equipo encargado especialmente de llevar adelante las pruebas?	x			
		25. ¿Existe un plan de pruebas definido?	x			
		26. ¿Cuál de los siguientes tipos de pruebas se realizan?		x		
		27. ¿Qué técnicas emplean para realizar las pruebas?		x		
		28. ¿De qué manera se ejecutan la mayor parte de las pruebas?	x			
		29. ¿Qué criterios emplean para dar por finalizado un ciclo de pruebas?		x		
		30. Observaciones acerca de las pruebas en entornos *Blockchain*				x
7	Mantenimiento	31. ¿Con que frecuencia realizan los siguientes tipos de mantenimiento?			x	
		32. ¿Cuál es el nivel de complejidad de realizar el mantenimiento?			x	
		33. Observaciones acerca del mantenimiento en entornos *Blockchain*				x
8	Gestión de la configuración	34. ¿Se documenta y realiza un seguimiento de los cambios en el proyecto?	x			
		35. ¿Cuál de las siguientes prácticas se emplean para el lanzamiento de una nueva versión del producto?				x
9	Gestión de Proyectos	36. Existe una planificación y gestión de proyectos definida (tiempos, recursos, costos…)	x			
		37. ¿Cuál de las siguientes prácticas se aplican en la planificación y gestión?				
		38. Observaciones acerca de la gestión de proyectos *Blockchain*				x
10	Aseguramiento de la Calidad	39. ¿Qué atributos de calidad consideran esenciales en el desarrollo *Blockchain*?		x		
		40. ¿Cuál de las siguientes prácticas de aseguramiento de la calidad se emplean?		x		

11	Otros	41. En breves palabras ¿Cuál considera que es el mayor desafió o complicación al desarrollar sobre la tecnología *Blockchain*?				x

2.2. Análisis de resultados.

La encuesta fue puesta a disposición de los miembros de 4 grupos de Argentina especializados en *Blockchain* y *Bitcoin* dentro de la red social Facebook durante el plazo de 30 días. Transcurrido ese tiempo se procedió a recuperar los datos para su análisis, con la sorpresa de que sólo se obtuvieron dos respuestas a la encuesta. Esta situación resulta sorpresiva si se consideran al menos tres aspectos que contextualizaron la distribución del instrumento: i) el tiempo en que estuvo disponible; ii) las 10 interacciones (comentarios y reacciones) que recibieron las publicaciones que contenían la motivación del trabajo de investigación y el link de acceso al cuestionario; iii) el alcance que tuvo la distribución del instrumento, que en sumatoria, en los cuatro grupos donde fue distribuida se contabiliza un total de 46.404 miembros, entre los que fue posible identificar algunos referentes y cabezas de compañía del sector *Blockchain y Bitcoin* del país.

Por otro lado, fue llamativo que la actividad en los grupos continuó constante posterior a la publicación y que las 2 respuestas estuvieron separadas en el tiempo por 27 días; esto lleva a suponer y descartar la posibilidad de que la encuesta no haya sido percibida por gran parte de los miembros en el período en que estuvo disponible. Otro factor que llamó la atención, en el contexto donde fue presentado el instrumento, es que un cuestionario perteneciente a una investigación académica que tenía a las criptomonedas y el comportamiento de sus usuarios como objeto de estudio, recibió

significativamente mayor atención en los grupos que el correspondiente al presente Trabajo Final de Graduación.

Una aproximación demográfica.

Se identifican dos profesionales de las ciencias de la computación, uno residente en la provincia de Buenos Aires (participante A), cuya trayectoria laboral en la industria del software lo posiciona como un perfil semi-senior (2 a 5 años), y otro perteneciente a la provincia de la Pampa (participante B) quien señaló contar con mayor antigüedad en la industria (más de 6 años).

En relación a la tecnología *Blockchain*, ambos encuestados declaran haberse formado de manera autodidacta, mientras que el participante bonaerense cuenta con menos de un año de experiencia trabajando en la misma, el participante pampeano, indicó estar involucrado con esta tecnología hace ya un periodo de entre 2 y 5 años.

En lo que respecta a los proyectos en los que se desenvuelven los encuestados, ambos tienen origen en Argentina y cuentan con su participación desde los inicios. La diferencia entre ambos participantes radica en el ámbito del proyecto, mientras que participante A indicó hacerlo por hobby, el participante B se definió como profesional en esta tecnología (trabaja o dirige una compañía de desarrollo).

Relacionado a las plataformas sobre las que trabajan los participantes, ambos coincidieron en desarrollar para Ethereum, mientras que, el participante B señalo ser partícipe además en proyectos apoyados en la red de Bitcoin y Cardano.

Proceso de desarrollo.

Adentrándose en el plano del proceso de software utilizado, el encuestado A señalo la ausencia de un marco que guíe el ciclo de vida durante el proyecto, aclarando, sin embargo, que en la fase donde centran mayormente la atención es en el desarrollo o codificación del sistema.

El encuestado B de lo contrario, señalo trabajar bajo la metodología Scrum durante el proyecto, coincidiendo con el participante A en que el mayor énfasis y esfuerzo durante el proceso está puesto en la fase de desarrollo.

Fase de requisitos.

Haciendo alusión a la fase de requisitos, el participante A señaló que las actividades relativas a esta etapa, como la captura y refinamiento, son prácticas que se realizan ocasionalmente y que los detalles de los mismos están presentes principalmente en la cabeza de los miembros del equipo, sin contar con un artefacto formal que los defina. En este sentido el encuestado se abstuvo de señalar un método por el cual eran especificados los requisitos y no emitió observación alguna acerca de esta etapa.

Por su parte el encuestado B señaló no realizar prácticas para la captura y refinamiento de los requisitos, sin embargo, esta respuesta se vuelve un tanto contradictoria al manifestar posteriormente mediante sus respuestas un enfoque más metódico durante esta fase que el primer encuestado. El participante B indicó que todos los miembros del equipo se mantienen bien informados sobre la estrategia y la dirección del producto, así como valerse de herramientas como User Stories para la representación

de los requisitos. Esto claramente se corresponde con la metodología indicada anteriormente por el participante (Scrum). Por otra parte, es de gran valor rescatar que, como observación sobre la etapa de requisitos, el encuestado comentó emplear la técnica de especificación formal para ciertos componentes core, que luego son validados con modelos y simulaciones.

Fase de análisis y diseño.

La fase de análisis y diseño fue señalada por el participante A como una etapa poco común en el proyecto, donde las prácticas asociadas no son ejecutadas a excepción de raras veces; el participante de la encuesta respondió no contar con un modelado del sistema, ni usar UML como lenguaje para tal fin. Otro factor de gran relevancia fue que apuntó no contemplar ningún atributo de calidad en esta etapa.

El encuestado B manifestó realizar prácticas de análisis y diseño de manera ocasional (un 2 en la escala Likert), para ello reportó valerse del Lenguaje Unificado de Modelado, centrando la documentación en modelos arquitectónicos y de comportamiento sobre la solución. Por otra parte, el encuestado manifestó la importancia de velar por los atributos de calidad del sistema durante el análisis y diseño, imputando uno de los valores máximos (3) en la escala Likert propuesta para este aspecto.

Fase de desarrollo o implementación.

El encuestado A indicó con respecto a esta fase que en el proyecto del cual es participe no se definieron estándares de codificación, siendo la única práctica para la

calidad empleada durante el desarrollo, la inspección de código / revisiones por pares.

En relación al participante B, su postura fue en gran parte opuesta a la de A. Indico que el equipo de desarrollo dispone de estándares de codificación definidos, además señaló que el equipo emplea diversas técnicas durante esta fase, entre las que seleccionó las inspecciones de código / revisiones por pares, análisis de código estático y documentación sistemática de código integrada.

Fase de pruebas.

La etapa de pruebas fue descripta por el participante A como una etapa informal, donde no existe un equipo especialmente dedicado a llevar adelante las mismas, ni un plan que guíe su ejecución. Se indicó que el único tipo de pruebas realizadas en el proyecto son de carácter unitario, que no emplean una herramienta para la automatización y que no utilizan ninguna de las técnicas específicas y bien conocidas para la elaboración de los casos de prueba. Por otro lado, se señaló que el criterio de finalización de las mismas está dado de manera arbitraria, cuando las pruebas unitarias se consideran suficientes, al no encontrarse más errores por parte del desarrollador, el ciclo concluye.

Desde la perspectiva del encuestado B, esta fase muestra mayor rigurosidad; indicó que en el proyecto tienen un equipo especialmente destinado a realizar las pruebas bajo una planificación bien definida. Además, señaló que se realizan pruebas de distinta índole a lo largo del ciclo de vida, como pruebas unitarias, de integración y de aceptación. Por otro lado, señaló no utilizar en el proyecto, técnicas específicas para el diseño de los casos de prueba, aunque si señaló contar con los mismos y realizar

inspecciones de código estáticas para validarlos, valiéndose de herramientas automatizadas para tal fin. Desde este enfoque más disciplinado que manifiesta el participante de la encuesta, indica que los ciclos de prueba son reportados como finalizados una vez que el 100% de los casos de prueba son ejecutados con éxito.

Fase de mantenimiento.

En lo vinculado a la fase de mantenimiento, el participante A, especificó que, a causa de la falta de documentación es una etapa prácticamente omitida, dejando de lado procesos adaptativos, correctivos y perfectivos sobre el sistema ya desplegado.

El participante B, señaló que, a pesar de la escasez de herramientas para soportar el mantenimiento sobre la Cadena de Bloques, se realizan los tres tipos de mantenimiento siendo los de mayor frecuencia los de tipo adaptativo (un 3 en la escala Likert), continuando por los correctivos (2 en la escala Likert), para finalmente atribuir un 1 en la escala Likert (raras veces) a los de tipo perfectivo.

Gestión de la configuración.

El encuestado A confirmó documentar el progreso, dejando entrever la existencia de artefactos donde los cambios son trazados. En relación a la liberación o puesta en producción de las versiones, el criterio empleado, es llegar a un acuerdo informal entre el grupo de desarrolladores del proyecto.

El encuestado B indicó que en el proyecto los cambios son documentados y trazados. Los responsables de definir las líneas base y de liberaciones son stakeholders

ajenos al equipo de desarrollo, por tal motivo, señaló además, que cuentan con entornos separados en lo que refiere a las pruebas y el ambiente de producción.

Gestión del proyecto.

Ampliando el enfoque hacia la gestión global del proyecto, A señaló que el grupo de trabajo no cuenta con una planificación definida del mismo, siendo la única práctica de gestión, la ejecución de reuniones de progreso semanales o mensuales.

El encuestado B, indicó que existe una planificación que contempla tiempos, recursos, costos y demás factores vinculados a la producción de software. Fiel a la metodología mencionada en un principio, realizan prácticas relativas a la gestión como planes operativos semanales, conocidos en la metodología Scrum como Sprints, planes para todo el proyecto en función del proceso definido (una aproximación a la gestión del Backlog según la metodología citada), informes de progreso y retrospectivas.

Aseguramiento de la calidad.

El aseguramiento de la calidad no está presente en el proyecto para el participante A, aunque indicó la importancia de contemplar la eficiencia (comportamiento en el tiempo, comportamiento en recursos) como atributo esencial de una solución desarrollada sobre esta tecnología.

El encuestado B, al respecto, señaló que considera como atributos de calidad fundamentales los agrupados bajo la categoría de funcionales (adecuación, exactitud,

seguridad, interoperabilidad) y que para velar por su cumplimiento emplean prácticas como revisiones de pares y auditorías sistemáticas.

Comentarios.

El encuestado A expresó que el mayor desafío o complicación a la hora de trabajar en un proyecto basado en la tecnológica *Blockchain* es la falta de estándares en cuanto al modelado de los sistemas. Mientras que el participante B se explayó describiendo que el mayor desafío al trabajar con esta tecnología se encuentra en la falta o inmadurez de las herramientas, desde el lenguaje, pasando por la documentación, compiladores, monitoreo, sistemas afines, entre otros.

2.3. Observaciones sobre la investigación de campo.

Si bien resulta imprudente realizar aseveraciones acerca del comportamiento de los desarrolladores *Blockchain* en Argentina en base a las 2 únicas respuestas recibidas, es posible, al sumar la interpretación y análisis del escenario que circundó la ejecución de la investigación, esbozar rasgos distintivos que dejan entrever el grado de madurez y enfoque con el que abordan dicha tecnología.

En primera instancia, resulta oportuno detenerse a evaluar las posibles causas de la baja participación de los desarrolladores para con el instrumento propuesto. Como se expuso a comienzos del apartado sobre el análisis de resultados, la contribución y predisposición de la comunidad de desarrolladores frente a la temática abordada por el presente Trabajo Final de Graduación, fue notoriamente baja. Ante esta situación, se optó

por analizar los perfiles de los individuos más activos dentro de los grupos objetivo y, particularmente, de aquellos quienes tuvieron alguna reacción hacia las publicaciones del cuestionario, con el fin de determinar los motivos que llevaron al parcial fracaso del instrumento. Un repaso por la biografía de estos individuos seleccionados, disponibles en la red social utilizada como medio de distribución de la encuesta, dejó en evidencia que la mayoría están involucrados, en diferentes medidas, con el desarrollo de software y/o la tecnología *Blockchain*; esta aproximación a los perfiles permitió arribar a dos presunciones; i) las personas que accedieron a la encuesta se encontraron con que no tienen el conocimiento suficiente para completarla; ii) los encuestados son capaces, pero no están motivados o simplemente no desean responder la encuesta.

Evaluando las conjeturas expuestas anteriormente, es posible dar cuenta que ambas acreditan de cierta forma la inferencia realizada sobre la marcada inmadurez en el medio; esta aseveración se justifica desde el argumento de que, si los posibles benefactores del presente Trabajo Final de Graduación no aprecian, contemplan o comprenden el valor del enfoque ingenieril sobre el que hace hincapié la investigación, se está ante un contexto poco profesional, similar al descripto en el apartado introductorio de la presente obra cuando se hizo referencia a la crisis del software. Este escenario se constituye en un principio como motivo suficiente para proceder bajo la premisa de que la situación de la comunidad de desarrolladores *Blockchain* en Argentina condice con lo expuesto durante el planteamiento del problema que motivo este Trabajo Final de Graduación.

Por otra parte, las respuestas obtenidas al cuestionario respaldan esta apreciación, ya que mediante el mismo, quedó en evidencia la carencia de prácticas específicas para el desarrollo sobre la Cadena de Bloques, circunstancia que como se pudo ver en el análisis de resultados, excede el perfil profesional de los encuestados.

Es meritorio, además, en este análisis final, distinguir los dos perfiles obtenidos de los encuestados ya que existe una palmaria brecha entre ellos. Por un lado, el participante A muestra mayor liviandad e informalidad general en los procesos durante el ciclo de vida del software, esto puede atribuirse a la corta experiencia tanto en su rol de desarrollador, como en la tecnología *Blockchain* en sí. Otro aspecto que indefectiblemente influye en sus respuestas es su compromiso con el proyecto, ya que como indicó en la encuesta, sólo lo hace por hobby. Por otro lado, el participante B, manifestó mayor antigüedad trabajando en la industria y puntualmente sobre la tecnología *Blockchain*, además señaló el compromiso laboral con el que se desenvuelve dentro del proyecto. Estos dos aspectos mencionados imprimieron rotundamente sus posteriores respuestas, donde fue notoria la intención y esfuerzo por implementar prácticas ya conocidas y metodologías populares para afrontar proyectos en la Cadena de Bloques, claramente esto manifiesta el acercamiento a una ingeniería de software con un enfoque sistemático, disciplinado y cuantificable. Sin embargo, mediante el comentario libre al final del cuestionario, dejó en claro que aún es necesario trabajar en la concepción de la ingeniería en software orientada a soluciones *Blockchain*.

Capítulo 3.

Antecedentes en ingeniería de software orientada a *Blockchain*

El campo de investigación de los métodos y prácticas de ingeniería de software dirigidas al desarrollo de software orientado a *Blockchain* (BOS) aún está en sus inicios. Sin embargo, al igual que el crecimiento exponencial que tuvo la tecnología *Blockchain*, las necesidades concretas y los aportes a un cuerpo de conocimiento que guíen la práctica sobre esta disciplina prosperan a un ritmo casi igual de vertiginoso.

El primer llamado a reflexionar sobre la necesidad de prácticas de la ingeniería en software orientadas a *Blockchain* viene de la mano del artículo de Porru et al. (2017) quienes abogan por el estudio y progreso de prácticas ingenieriles sólidas que garanticen actividades de prueba efectivas, mejoren la colaboración en equipos grandes y faciliten el desarrollo de contratos inteligentes. Puntualmente en su artículo hacen alusión a los siguientes desafíos: la necesidad de nuevos roles profesionales, prácticas transversales al ciclo de vida de software que garanticen la seguridad y confiabilidad de los productos resultantes, notaciones específicas, patrones y metamodelos arquitectónicos, y métricas concretas para el control del proceso.

3.1. Contribuciones al ciclo de vida de desarrollo de software *Blockchain*

Partiendo de los desafíos mencionados en el artículo Blockchain-oriented Software Engineering: Challenges and New Directions (Porru, Pinna, Michele, & Tonelli, 2017) y las principales actividades del ciclo de vida del software, se sistematizó y ordenó la búsqueda de obras con rigurosidad académica que aborden los diferentes aspectos en

vistas de realizar una ligera aproximación al estado de arte sobre los conocimientos y prácticas ingenieriles específicas para esta tecnología.

Tomando como punto inicial la etapa de requisitos, es importante tener en cuenta que el principal factor distintivo, en comparación con la elicitación realizada para el software tradicional, recae en analizar si las necesidades del sistema realmente son meritorias de emplear la tecnología *Blockchain* y en evaluar cómo el mismo puede verse beneficiado por ella. Bajo este enfoque el NIST a través del reporte de Yaga, Mell, Roby, & Scarfone (2018), destina un capítulo entero (capitulo 8) a tratar las consideraciones y escenarios para el análisis que permitan determinar la idoneidad de esta tecnología en un proyecto. En dicho capítulo, citando a The United States Department of Homeland Security (DHS) Science & Technology Directorate, los autores presentan un diagrama de flujo que permite de manera sencilla arribar a la noción de si la Cadena de Bloques es la alternativa indicada para un determinado proyecto de desarrollo. Además, siguiendo con esta línea, resultan de gran relevancia las referencias, introducidas en el reporte, acerca de los diversos trabajos académicos que abordan mediante metodologías y herramientas el interrogante ¿Necesito una *Blockchain*?

Por otra parte, en el marco de esta fase, es también posible considerar el artículo *A Solution in Search of a Problem: A Method for the Development of Blockchain Use,* propuesto por Fridgen et al. (2018), donde mediante un enfoque de diseño de investigación para la acción e ingeniería de métodos situacionales, propone un procedimiento para el desarrollo de Casos de uso *Blockchain*. Sus autores además acompañan su propuesta exponiendo la evaluación exitosa de la misma en cuatro

diferentes industrias: banca, seguros, construcción y automotriz. A diferencia de la perspectiva del reporte de Yaga et al., donde se analiza la adecuación de esta tecnología a un escenario existente, el enfoque propuesto por esta obra parte de las cualidades de la tecnología *Blockchain* para proceder sistemáticamente con el desarrollo de nuevos casos de uso en una industria especifica.

Al hablar de esta etapa de elicitación, es donde también se puede ubicar la necesidad de nuevos roles profesionales expresada por Porru et al., ya que las implicancias significativas que tiene *Blockchain* en el marco legal, organizacional y económico, requieren una persona formada en estas áreas y que a su vez cuente con el claro entendimiento sobre las posibilidades y limitaciones de la tecnología *Blockchain*. Al respecto existen propuestas educativas y material bibliográfico que ofrecen preparación sobre *Blockchain* sumamente útil para ejecutivos de organizaciones, por lo cual éste es un aspecto que ya tiene un rumbo definido y se espera madure con el tiempo y las necesidades específicas presentes en las diferentes industrias donde *Blockchain* gane terreno.

Haciendo alusión a la fase de análisis y diseño, uno de los criterios más comprometidos acorde a los desarrolladores encuestados, existen valiosas obras que ofrecen una aproximación a métodos y herramientas específicas para el modelado de soluciones *Blockchain*.

Xu et al. (2017) propone una taxonomía de conceptos relativos a la Cadena de Bloques que permite ayudar con el diseño y la evaluación de su impacto en las arquitecturas de software. Este trabajo puede ser visto como la articulación entre los

requerimientos y el diseño arquitectónico de una solución. La taxonomía ofrecida por los autores captura los principales aspectos arquitectónicos de la tecnología *Blockchain* y reflexiona sobre el impacto de sus principales decisiones de diseño. Para llevar a la práctica dicha clasificación Xu et al. presenta un diagrama de flujo y una lista de verificación inicial que orientan las decisiones de diseño ayudando con importantes consideraciones arquitectónicas que repercuten sobre los atributos de rendimiento y calidad de los sistemas basados en esta tecnología.

Por su parte, Wessling et al. propone un enfoque para decidir qué elementos de la arquitectura de una aplicación podrían beneficiarse del uso de la tecnología *Blockchain*. El método propuesto parte de identificar a los participantes, sus relaciones de confianza e interacciones para derivar una arquitectura que integra la tecnología *Blockchain* en los sistemas de software existentes o para la creación de nuevos sistemas que utilicen *Blockchain* sólo en ciertas partes.

Con respecto a las posibles extensiones al Lenguaje de modelado unificado, se han publicado varios artículos que proponen ampliaciones de UML para dar soporte a la representación en campos específicos de aplicación.

Aunque puntualmente para la tecnología *Blockchain* no existe demasiado material, el enfoque de trabajos como el propuesto por Baumeister et al. (1999) o el correspondiente a Baresi, Garzotto & Paolin (2001), donde, para dar respuesta al modelado de sistemas web e hipermedia, presentan nuevos estereotipos y estructuras de navegación que se combinan con primitivas funcionales y de comportamiento

proporcionados por UML, permiten conocer y hacer propios mecanismos para adaptar y extender el lenguaje de modelado a las necesidades particulares del sistema.

Por su parte y mediante una aproximación más específica a los propósitos del objeto de estudio de este trabajo, Rocha y Ducasse (2018), en un artículo reciente, desarrollan y exponen tres enfoques de modelado complementario, basados en modelos bien conocidos de ingeniería de software (diagramas E-R, UML y BPMN), para luego aplicarlos a un ejemplo de diseño de software orientado a *Blockchain*.

Al indagar en la fase de implementación, se encontró que plataformas maduras que soportan el despliegue de DApps, como es el caso de Ethereum, disponen de una vasta documentación mantenida por la comunidad de desarrolladores[16]. Esta documentación de libre acceso presenta desde las particularidades técnicas y arquitectónicas de la plataforma, patrones comunes en contratos inteligentes, hasta buenas prácticas que conducen al desarrollo seguro[17].

Por su parte el lenguaje ampliamente utilizado en dicha plataforma, Solidity, también dispone de documentación oficial que abarca además de la sintaxis, patrones de diseño comunes, guías para la especificación, reporte de errores, entre otras temáticas relativas a la utilización del lenguaje.[18]

---

[16] La wiki desarrollada y mantenida por la comunidad puede encontrarse en el enlace: https://github.com/ethereum/wiki/wiki/Ethereum-Development-Tutorial.

[17] Parte de la wiki de Ethereum mantenida por ConsenSys dedicada a las buenas prácticas de desarrollo, disponible en https://consensys.github.io/smart-contract-best-practices/

[18] Documentación disponible en https://solidity.readthedocs.io/en/develop/

Por fuera de la mencionada documentación “oficial” de la plataforma y el lenguaje, es posible encontrar trabajos como el de Massimo Bartoletti & Livio Pompianu (2017) donde, tras analizar y definir una taxonomía de *Smart Contracts* en base a un estudio de 811 proyectos, proponen patrones de diseño para contratos inteligentes, a los cuales luego correlaciona con el dominio de aplicación, dando soporte de esta forma a la elección del diseño más apropiado.

Maximilian Wöhrer & Uwe Zdun (2018) presentaron dos trabajos sobre patrones de diseño en el desarrollo *Blockchain*, por un lado, el titulado Design Patterns for *Smart Contracts* in the Ethereum Ecosystem, ofrece la descripción de patrones de diseño que fijan pautas en el desarrollo de contratos inteligentes sobre la plataforma Ethereum. Los autores explican en detalle y proporcionan ejemplos de patrones que responden a necesidades comunes en el diseño de *Smart Contracts* durante el abordaje de los requisitos de una aplicación y permiten resolver problemas comunes. Por otro lado, el trabajo llamado *Smart Contracts*: Security Patterns in the Ethereum Ecosystem and Solidity, presenta un enfoque específico sobre los patrones de diseño desde la mirada puesta en la seguridad de los contratos inteligentes. Maximilian Wöhrer & Uwe Zdun (2018) detallan soluciones a problemas de seguridad típicos con el fin de que los desarrolladores Solidity puedan aplicarlos para conseguir mitigar los posibles escenarios de ataque.

Kees Boogaard (2018) propone la utilización del enfoque de desarrollo dirigido por modelos para el diseño e implementación de los *Smart Contracts*, en su tesis de maestría presenta como es posible en base a la ingeniería de modelos generar un proceso

que permite cerrar la brecha semántica entre el conocimiento del dominio y el contrato inteligente a la vez que ayuda a los desarrolladores a crear contratos inteligentes menos vulnerables que representen el problema con mayor precisión.

El trabajo de Kees Boogard es absolutamente valioso y aplicable, al proponer con completitud un proceso sostenido por prácticas ingenieriles para el diseño y desarrollo de Contratos Inteligentes. Nuestro posterior modelo sugiere fuertemente emplear esta aproximación dentro de una de sus actividades estructurales.

Desde una perspectiva transversal al desarrollo, existen también aportes como el de Tonelli, Destefanis, Marchesi, & Ortu (2018) que estudian y proponen métricas específicas para aplicar en la implementación de *Smart Contracts*.

En relación a las pruebas Ashray Kakadiya (2017) realiza un importante trabajo de investigación buscando y referenciando antecedentes sobre herramientas y métodos referidos a esta etapa, para luego, al notar la ausencia de un enfoque similar, desarrollar una metodología de pruebas que acompaña en su totalidad el ciclo de vida del software orientado a *Blockchain*. Entre los artículos citados por Kakadiya cabe mencionar enfoques que van desde la proposición de diferentes *frameworks* para probar soluciones en Ethereum, *Smart Contracts* testeados mediante otros *Smart Contracts*, o inclusive trabajos de investigación donde mediante un caso de uso proponen un modelo para realizar pruebas de verificación utilizando la cadena de Markov. La cadena de Markov, es un enfoque ampliamente reconocido para garantizar la corrección de un sistema al verificar que cualquiera de sus comportamientos es un modelo para una propiedad determinada.

Por su parte, y también citado por Kakadiya, Infosys (2018), una empresa multinacional de servicios de tecnologías de la información con base en Bangalore, India, emitió un documento técnico titulado Assuring success in *Blockchain* implementations by engineering quality in validation. En este documento, discuten varios desafíos en la prueba de implementaciones de *Blockchain* y enumeran las fases de prueba con el volumen de pruebas, metodología y herramientas para llevarlas a cabo. También en su artículo, discuten la estrategia de prueba a través de varias fases del ciclo de vida, apuntando las actividades clave de cada fase.

En lo referido al mantenimiento, no se encontraron estrategias ni herramientas específicas para la Cadena de Bloques durante esta etapa, aunque mediante los avances expuestos con anterioridad es posible documentar y estructurar con un desarrollo *Blockchain* con el fin de facilitar las tareas correspondientes al mantenimiento de la aplicación en todas sus formas.

Concluyendo esta sección de antecedentes, es de mayor importancia para el presente Trabajo Final de Graduación, referenciar tanto la propuesta de Marchesi, Marchesi, & Tonelli (2018), An Agile Software Engineering Method to Design Blockchain Applications, donde proponen una metodología basada en principios ágiles para el desarrollo de software orientado a *Blockchain*; como el enfoque ofrecido por Almeida, Albuquerque, & Silva (2018), sobre un proceso de desarrollo de software que sienta sus bases en la metodología lean startup, donde se suguieren además de las actividades y su flujo, artefactos, métricas y roles.

Del trabajo realizado por Marchesi et al. se recuperan múltiples ideas acerca de un posible enfoque metodológico, aunque en vistas a lo ya abordado durante esta investigación, no se acuerda con la postura de que la agilidad en su estado puro resulte idónea para un proceso orientado a *Blockchain*. En tanto, la aproximación propuesta por Almeida et al. resulta enriquecedora para el modelo desarrollado en el siguiente apartado si se tiene en cuenta la completitud del proceso especificado a travez del papel que cumplen los artefactos, métricas y la visión paralela sobre el proceso que afecta al grupo de desarrollo y al encargado de gestionar el proyecto, aunque su enfoque pensado para startups compromete en cierta mediada el uso de prácticas que garanticen un software seguro y una gestión apropiada de los recursos.

3.2. Observaciones finales sobre los antecedentes hallados.

Es notorio como en torno al artículo de Poruu et al. (2017) comenzaron a proliferar los trabajos académicos y el interés de los profesionales del software en disciplinar el desarrollo *Blockcahin.* Los aportes realizados desde entonces, abarcan aspectos pertenecientes a las diferentes fases del ciclo de vida de desarrollo, presentando propuestas concretas acerca de cómo valerse de técnicas, métodos y herramientas que posibiliten la construcción de sistemas *Blockchain* fundados en el análisis y diseño previos, así como también en prácticas para las etapas sucesivas.

Por otra parte, cabe remarcar que un aspecto de importancia y aún no resuelto en su totalidad, es el lenguaje de modelado utilizado para el diseño en dicha tecnología. Como se mencionó en este capítulo, al respecto se retoman ideas que se gestaron hace

casi 20 años cuando el mundo del software se enfrentaba al auge de los sistemas web e hipermedia. Si bien estas propuestas son fuertes cimientos para derivar adaptaciones propias para los sistemas *Blockchain*, la falta de especificidad y particularidades de la tecnología, convierten esta tarea en un desafío en sí mismo. Ideas más acabadas con respecto a *Blockchain* y su modelado aparecieron recientemente, sin embargo, siguen estando libradas a la interpretación y conveniencia de quienes optan por aplicarlas a sus proyectos.

En lo concerniente al resto de las fases del ciclo de vida de desarrollo se encontró material de sumo valor y completitud, aunque en la mayoría de los casos, la documentación, se ciñe específicamente a plataformas e implementaciones particulares de esta tecnología.

Acorde a lo expuesto, es coherente concluir con la idea de que la falta de estandarización en los conceptos y la manera de implementar *Blockchain* en un sistema, son el mayor obstáculo a la hora de desplegar prácticas y herramientas de las que pueda valerse un equipo de desarrollo. Es esta falta de estandarización la que genera una brecha y desnormaliza la actividad de los profesionales del software avocados a esta tecnología, forzándolos a transitar por un proceso desordenado y abierto a la improvisación.

Capítulo 4.

Un modelo de procesos para el desarrollo *Blockchain*

Partiendo de la comprensión de los fundamentos y particularidades que hacen único al desarrollo de software sobre la tecnología *Blockchain*, sumado a la carencia de buenas prácticas y un marco que guíe el proceso evidenciados durante el estudio de campo y documental previos; el objetivo de este cuarto capítulo es propone un modelo de proceso enfocado en proyectos de software orientado a la Cadena de Bloques.

4.1. Diferencia entre proceso y modelo de proceso de Software

Resulta importante para interpretar la propuesta resultante de la investigación, diferenciar los conceptos de procesos de software y modelos de procesos de software. Ambos términos son usados generalmente de manera indistinta, aunque técnicamente persiguen objetivos y granularidades disímiles.

Los procesos de desarrollo de software son “una serie de actividades relacionadas que conducen a la elaboración de un producto de software” (Sommerville, 2011).

Partiendo de esta definición resulta notoria la libertad y arbitrariedad con la que es posible generar un proceso que se ajuste a un proyecto en particular, sin embargo, Pressman (2010) propone un enfoque basado en componentes en el cual jerarquiza el trabajo técnico involucrado, delimitando piezas esenciales que guían a la completitud y promueven el éxito de un proceso en un proyecto ingenieril de desarrollo de software.

Para Pressman lo primero a ser tenido en cuenta a la hora de describir un proceso de software es la definición de una estructura, la misma está compuesta por un conjunto

reducido de actividades aplicables a todos los proyectos, sin importar su tamaño o complejidad; la estructura de un proceso de software es también conocida como ciclo de vida del desarrollo de software. Somerville (2011), al respecto, propone una representación de los procesos de software como se observa en la figura 6, definiendo una estructura de 4 actividades comunes a cualquier proceso de software: especificación, diseño e implementación, validación y evaluación. Pressman, por su parte, destaca 5 actividades estructurales genéricas: comunicación, planeación, modelado, construcción y despliegue.

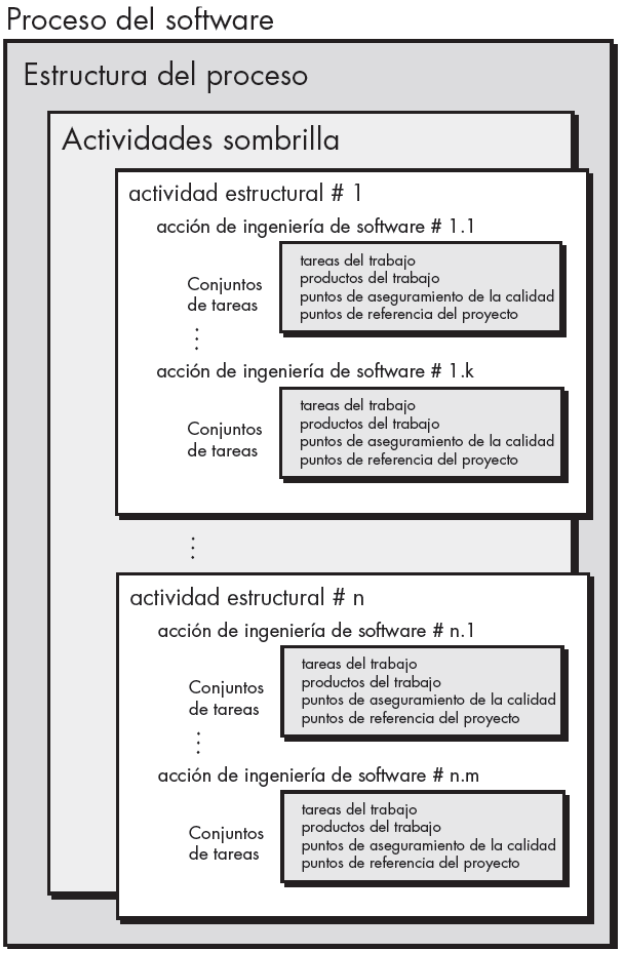

Figura 6. Esquema de un proceso de software según Pressman.
Imagen recuperada de (Pressman, 2010, pág. 27)

Continuando con el esquema, se percibe que por debajo de la estructura se encuentra un bloque de actividades sombrillas que acompañan a las actividades estructurales a lo largo de todo el proceso. Entre ellas se pueden mencionar las relacionadas al seguimiento y control del proyecto, administración de riesgos, aseguramiento de la calidad, administración de la configuración, entre otras.

Volviendo sobre las actividades estructurales, éstas, están a su vez formadas por un conjunto de acciones de la ingeniería de software, estas acciones se encuentran definidas por un grupo de tareas de trabajo, los productos resultantes de las mismas, los puntos de aseguramiento de la calidad y los puntos de control de avance (Pressman, 2010).

Por su parte, otro aspecto que resulta fundamental a la hora de pensar en un proceso de desarrollo de software y que involucra a los componentes anteriores, es el orden en el que se organizan y ejecutan las actividades estructurales con respecto a la secuencia y el tiempo. Esta característica es conocida como flujo del proceso, y para Pressman (2010) es posible distinguir, como se aprecia en la figura 7, principalmente cuatro alternativas: a) un flujo de proceso lineal donde se ejecuta cada una de las actividades estructurales en secuencia; b) un flujo de proceso iterativo en el cual repite una o más de las actividades antes de pasar a la siguiente; c) un flujo de proceso evolutivo donde se realizan las actividades en forma “circular” y cada circuito lleva a una versión más completa del software; d) un flujo de proceso paralelo en el que se ejecuta una o más actividades en paralelo con otras. Es importante aclarar que estos flujos no son

mutuamente excluyentes y es posible combinarlos en la definición de un proceso para el desarrollo de software.

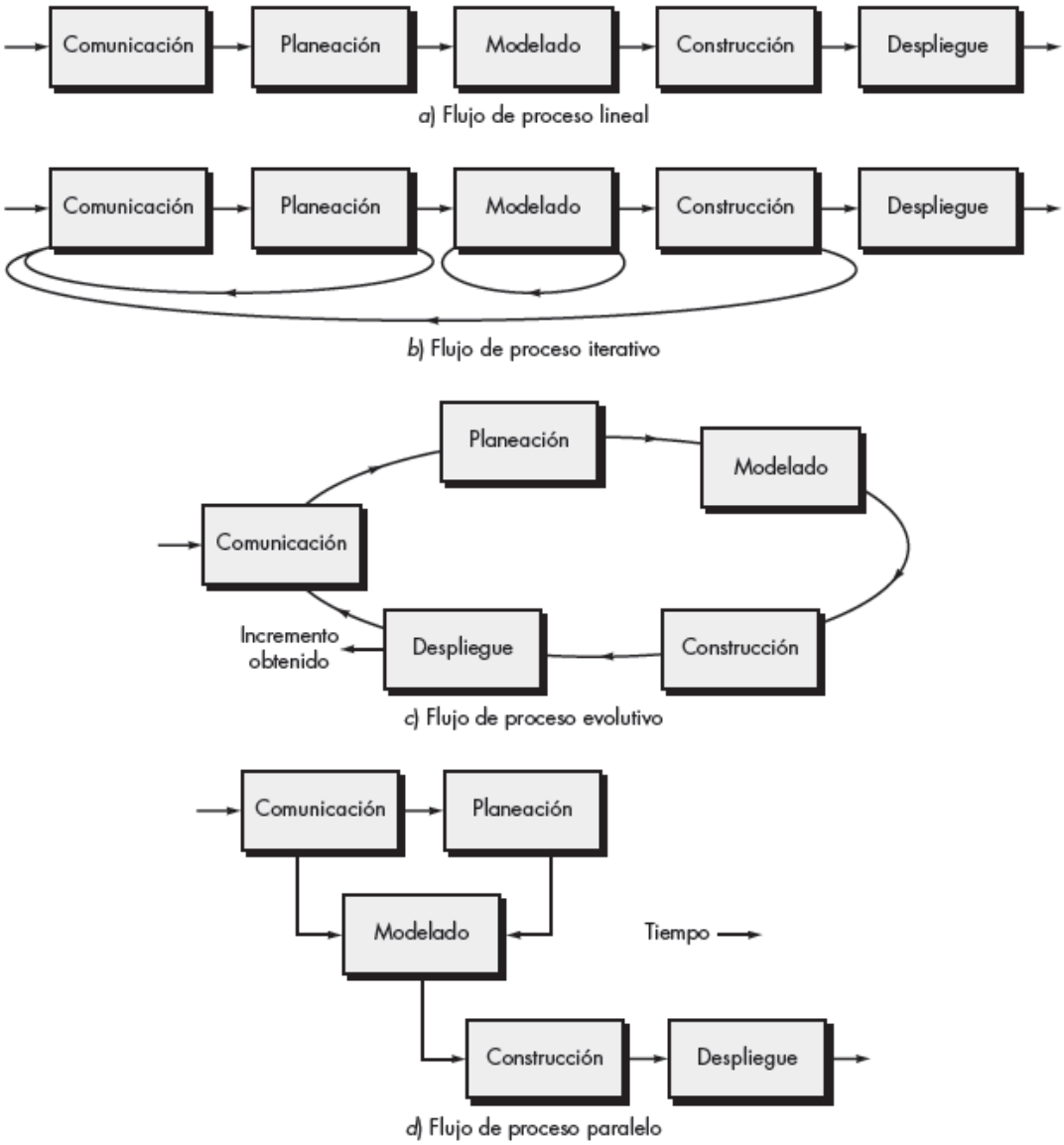

Figura 7. Tipos de flujo de proceso.

Imagen recuperada de (Pressman, 2010, pág. 28)

Finalmente, existe una clasificación generalizada de los procesos según su comportamiento acorde avanza el proyecto, es posible diferenciar los procesos de software dirigidos por un plan de los procesos ágiles. Los procesos dirigidos por un plan son aquellos donde todas las actividades del proceso se planean por anticipado y el avance se mide contra dicho plan. En los procesos ágiles, la planeación es incremental y se ajusta con facilidad en respuesta a los requerimientos cambiantes del cliente que puedan surgir durante el desarrollo del proyecto.

En cuanto al concepto de modelo de procesos, es posible definirlo como una representación abstracta y simplificada de un proceso de software bajo una perspectiva en particular. Cada modelo, ofrece sólo información parcial acerca de dicho proceso bajo un enfoque específico abordado por el mismo (Sommerville, 2011).

Los modelos de proceso tienen carácter prescriptivo, es decir, definen un conjunto prescrito de elementos del proceso y un flujo predecible para la ejecución del mismo.

Todos los modelos de proceso de software pueden incluir las actividades estructurales generales mencionadas con anterioridad, aunque es fundamental en su diferenciación que cada uno ponga distinto énfasis en ellas y en la forma particular del flujo que invoca cada actividad (Pressman, 2010). En otras palabras, las principales variables que definen un modelo de procesos, son las actividades estructurales y el flujo de ejecución de las mismas.

4.2. Propuesta MDSOB

Partiendo de las diferencias existentes entre el objetivo de un proceso de desarrollo de software y un modelo de procesos, se optó por este último como propuesta del presente Trabajo Final de Graduación. La elección de un modelo de proceso responde al grado de abstracción necesarios para hacer frente al nuevo paradigma tecnológico que envuelve la Cadena de Bloques; como se expuso en el primer capítulo, existen diversas implementaciones *Blockchain* las cuales responden a determinadas particularidades, mediante un modelo, es posible desentenderse de estas diferencias sin que el mismo pierda valor y utilidad como un framework adaptable a las necesidades del proyecto. Esta propuesta se posiciona desde una perspectiva arquitectónica, es decir, expone un marco de trabajo del proceso, sin precisar detalles de las actividades, roles y tareas específicas que conforman atómicamente dicha estructura. A pesar de que prime este enfoque en la propuesta, se sugerirán aspectos que guíen la ejecución de cada una de las fases a modo de enriquecer y facilitar la aplicabilidad del marco ofrecido.

Como fue mencionado con anterioridad, cuando se trató la noción de modelo de procesos, el valor distintivo de esta propuesta recae en la definición de nuevas fases o actividades estructurales y en el flujo particular que guía la ejecución de las mismas.

El modelo para el desarrollo de software orientado a *Blockchain* (desde ahora MDSOB) es un modelo que, ante la incapacidad de los modelos existentes de responder de manera integra a las particularidades de la tecnología *Blockchain* en el desarrollo de software, ofrece una alternativa especialmente pensada para el abordaje del mismo. MDSOB se sustenta en principios generales de la ingeniería en software y en parte de los

enfoques propuestos por modelos de desarrollo ya conocidos, entre los que se pueden mencionar aportes del desarrollo seguro, modelo en cascada, modelo iterativo incremental y metodologías ágiles.

El MDSOB es un modelo que combina un enfoque basado en planificación con un enfoque ágil, la coexistencia de ambos bajo un mismo modelo es posible debido a la descomposición sugerida en el capítulo 1 cuando se trató la temática referida a las DApps y su arquitectura a alto nivel. La separación de la aplicación cliente de la capa donde interviene la tecnología *Blockchain*, permite tratar la primera mediante un enfoque ágil iterativo e incremental basada en el desarrollo por prototipos, lo que conduce a una pronta aproximación al producto, mientras en simultaneo se trabaja bajo un enfoque planificado y seguro, sobre los aspectos sensibles vinculados a la tecnología *Blockchain.* A raíz de esta separación, el flujo de procesos del MDSOB se ve afectado y representado por una combinación de las tipologías expuestas en la sección 5.1, así, las actividades estructurales son iniciadas en una secuencia lineal para luego bifurcarse en un flujo paralelo que distingue un camino evolutivo, de uno secuencial para finalmente volverse a integrar.

Desde la perspectiva del desarrollo seguro, el MDSOB se posiciona sobre la metodología Microsoft Security Development Lifecycle, siendo completamente viable la adopción de las practicas propuestas por dicho framework a las distintas fases del modelo. Por otro lado, se recomienda acompañar el ciclo con las prácticas sugeridas en el sitio Ethereum Smart Contract Best Practices, las cuales son aceptadas como referentes

dentro del ecosistema de desarrolladores Ethereum y sumamente extrapolable a otras plataformas *Blockchain*.

MDSOB se divide en 5 fases o actividades estructurales fundamentales, las 2 primeras fases deben desarrollarse bajo un enfoque planificado, bien definido y secuencial, ya que el resultante de ellas será el puntapié inicial para que el equipo de desarrollo pueda avocarse a los dos planos principales necesarios para construir un sistema en base a esta tecnología. Las fases 3 y 4 son ejecutadas en simultaneo estando compuestas por procesos independientes autocontenidos; mientras que en la fase 3 se sugiere un proceso más rígido y documentado que mitigue al máximo las posibles fallas del sistema al ser desplegado el Smart Contract en la Cadena de Bloques, la fase 4 mantiene un espíritu ágil, inspirado en el desarrollo por prototipos. Esta división posibilita el trabajo en paralelo sobre diferentes intereses del sistema a la vez que aumentan los puntos de control y la flexibilidad de adaptación a proyectos diferentes. Además, mediante esta división en el modelo, se facilita la gestión de tiempos en el proyecto, compensando las múltiples y necesarias interacciones del desarrollo de la aplicación cliente para su validación por parte de los stakeholders, con el proceso estructurado del equipo abocado a la Cadena de Bloques quienes tienen un enfoque secuencial y más hermético posterior a que el diseño del Smart Contract fuera validado. Esta gestión permite que ambos equipos puedan mantener su propia planificación acorde a sus prioridades, procurando compartir una fecha de integración común.

Las 5 actividades estructurales propuestas en el MDSOB son Elicitación, Especificación, Desarrollo *Blockchain*- Smart Contract, Desarrollo aplicación cliente,

Integración y despliegue. Como puede notarse en sus nombres, las etapas, a diferencia de otros modelos, no reflejan directamente las actividades del ciclo de vida de software genérico, sino que están pensadas en la agrupación de actividades acordes a las responsabilidades y productos a los que conducen. Por lo tanto, para su interpretación cada una de ellas será explicada desde sus objetivos, actividades esenciales (sin ser prescriptivo), recomendaciones oportunas y medidas de seguridad sugeridas para acompañar dichas actividades.

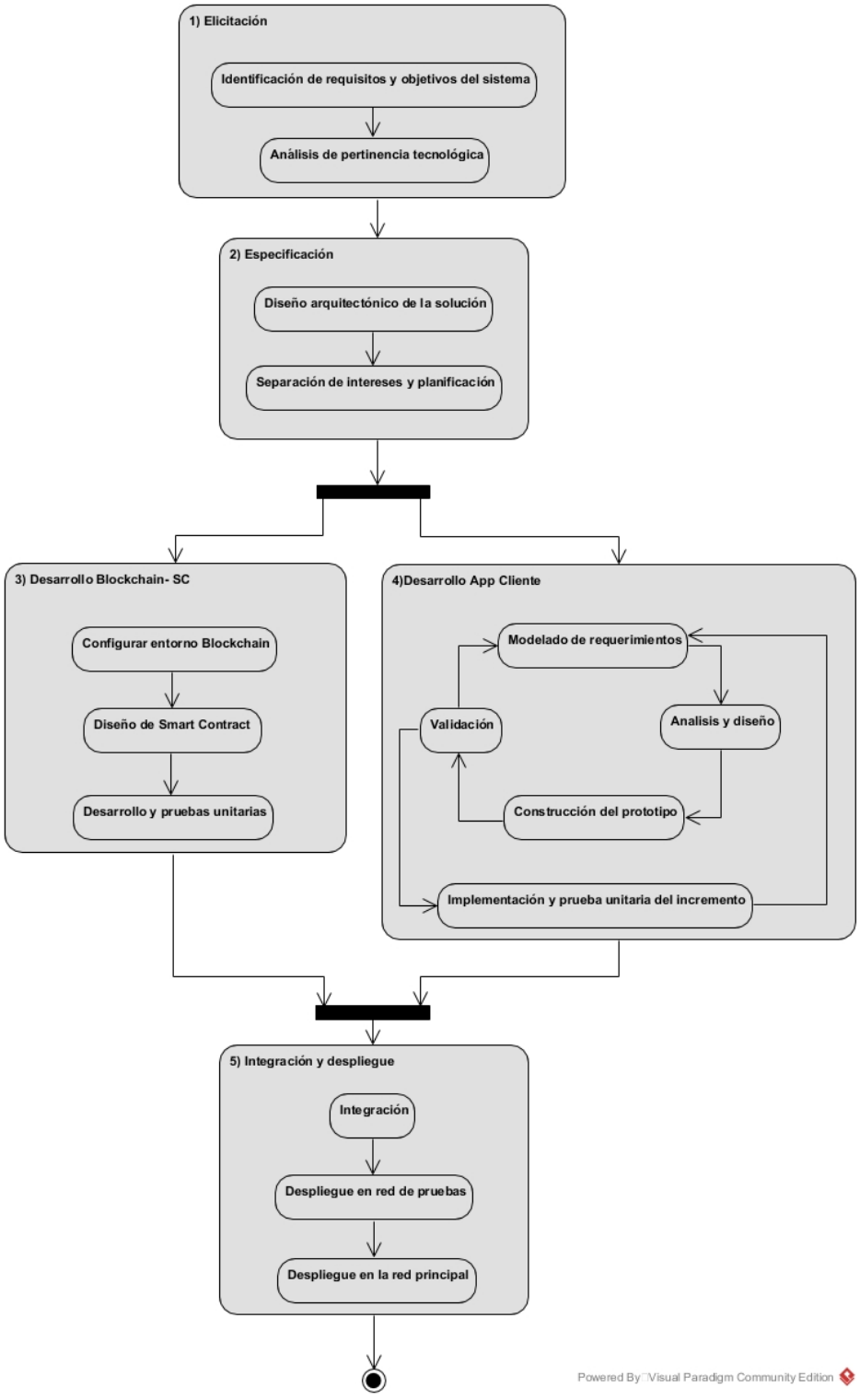

Figura 8. Diagrama arquitectónico del MDSOB.

Elicitación: Esta fase está caracterizada por comprender el modelo de negocios y las necesidades del sistema. Entre las actividades fundamentales de esta fase se encuentran por un lado la definición de los servicios, las restricciones y las metas del sistema; para ello se recomienda trabajar en estrecha relación con los interesados, tratando de capturar al máximo detalle los requisitos. Es importante para lograr una correcta aproximación en las actividades siguientes relevar aspectos como: los participantes de la red, diferenciar entes reguladores y entidades reguladas, determinar la gobernanza de la red, las formas de monitoreo, los controles de acceso; los flujos transaccionales y la procedencia de los activos digitales, el tipo y necesidades de los datos a registrarse; requisitos de seguridad criptográfica; procesos donde se requiera reducir o eliminar los esfuerzos manuales de reconciliación y resolución de conflictos. Todos estos aspectos deberían quedar documentados y detallados ya que tienen repercusión en la posterior arquitectura de la solución.

Partiendo de los requisitos capturados, se debe someter los mismos a alguno de los *frameworks* apuntados con anterioridad para analizar que la tecnología *Blockchain* sea realmente la solución indicada, esta actividad se ve representada por el análisis de pertinencia tecnológica en la figura 8. Por su naturaleza sencilla y la orientación a otras posibles soluciones distintas a la Cadena de Bloques, se recomienda el flujograma propuesto por el informe del NIST.

En lo que respecta a la seguridad en esta etapa es importante considerar concienzudamente los requisitos de privacidad y seguridad del sistema, partiendo de ellos hacer un temprano análisis de los riesgos y su evaluación en caso de ocurrencia, con esta

base definir también los mínimos aceptables en relación a estos aspectos. Contemplar de manera temprana dichos atributos y dejarlos explícitamente documentados ayuda al equipo a comprender los riesgos asociados, identificar y solucionar errores de seguridad durante el desarrollo, algo de suma importancia al tratar con esta tecnología.

Especificación: Esta etapa se nutre de la anterior, tras un análisis de los requerimientos generales capturados durante la elicitación se procede a modelar el sistema desde un enfoque global a través de las vistas que se consideren necesarias. Para tal fin es recomendable y de suma utilidad valerse de la taxonomía y método propuesto por Xu et al. (2017) que permite hacer una traza casi directa entre los requisitos capturados y las necesidades arquitectónicas de una Cadena de Bloques.

Teniendo como origen este primer modelo del sistema, ambos equipos de desarrollo deben definir las responsabilidades y trabajo pertinentes para cada uno de ellos, es aquí donde se realiza una planificación del proyecto en conjunto, fijando fechas que involucren a todo el grupo de desarrolladores siendo estas fechas de integración, reportes de avance y puestas en común para mantener en máxima sincronía el trabajo de ambos grupos. Claramente el rol del Project manager juega un papel fundamental durante esta etapa ya que es un desafío mantener en sincronía equipos con intereses dispares.

Desde la perspectiva de la seguridad, esta primera aproximación técnica a la solución permitirá revisar y ajustar el artefacto destinado a esta faceta elaborado en la etapa anterior.

Desarrollo Smart Contract- Blockchain: Como su nombre lo indica esta etapa está enfocada en el desarrollo y configuración de los componentes de la tecnología *Blockchain*, así como en el diseño e implementación de los *Smart Contracts* necesarios para ejecutar la lógica de negocio elicitada y modelada en etapas anteriores.

Como primera actividad dentro de esta fase se recomienda la preparación del entorno *Blockchain*, con esto no sólo se hace referencia a seleccionar el set de herramientas para llevar adelante el desarrollo y las pruebas, sino contemplar cuestiones asociadas a la infraestructura. Entre los pasos a seguir se sugiere: identificar el mecanismo de consenso más adecuado, identificar la plataforma apropiada que de soporte al mecanismo seleccionado, diseñar el entorno de ejecución y definir responsabilidades de los nodos de la red, ajustar los parámetros de la instancia considerando la gestión de claves, las firmas, parámetros y recursos, formatos de direcciones y de claves, entre otros.

Una vez que la infraestructura tecnológica este definida y configurada, es necesario tener en consideración las interfaces de programación de aplicaciones (API) necesarias para interactuar con la Cadena de Bloques, se pueden mencionar a grandes rasgos las interfases para generar pares de claves y direcciones, realizar funciones relacionadas con la auditoría, autenticación de datos a través de firmas digitales y hashes, entre otras. Es importante remarcar que las plataformas más maduras y usadas en el ecosistema *Blockchain*, ya cuentan con este conjunto de subrutinas disponibles para su uso.

En lo que respecta a los Contratos inteligentes indefectiblemente por sus características de inmutabilidad se recomienda llevar adelante un proceso planificado y

bien documentado, aunque por el dinamismo de los actuales proyectos un enfoque iterativo incremental siempre será una buena opción si la disgregación del problema lo permite. Sin embargo dependiendo la complejidad y exclusividad del contrato a ser desarrollado los enfoques posibles son diversos; desde la reutilización mediante el uso de patrones ya conocidos y probados que dan solución a requisitos recurrentes como los propuestos por Maximilian Wöhrer & Uwe Zdun (2018) o Massimo Bartoletti & Livio Pompianu (2017), pasando por un enfoque a medida acorde al modelo de negocio a través del proceso basado en la ingeniería dirigida por modelos propuesto por Kees Boogaard (2018), hasta un enfoque de desarrollo de sala limpia y el uso de métodos formales si la criticidad del desarrollo lo amerita. La recomendación en este punto es que, si existe un diseño ya verificado por la comunidad y desplegado en otros servicios de la Cadena de Bloques, decantarse por esta alternativa garantizara desde un principio mayor grado de seguridad, mayor soporte y menor esfuerzo en su implementación.

Al momento de la implementación, es necesario tener una mentalidad preparada para el fracaso, cualquier contrato inteligente no trivial presentara errores, por lo tanto, es importante tomar recaudos como asegurar que la lógica del contrato sea simple (a mayor complejidad aumentan las probabilidades de errores), modularizar el código para mantener pequeños contratos y funciones, y priorizar la claridad del código frente al rendimiento.

Un sistema de contratos inteligentes desde la perspectiva de la ingeniería de software debe ser modular, reusar componentes en lugar de duplicarlos y soportar componentes actualizables, bajo una mirada arquitectónica estas propiedades también

deberían respetarse. Sin embargo, hay excepciones importantes donde las mejores prácticas de seguridad e ingeniería de software pueden no estar alineadas. Es aquí donde un análisis de las dimensiones que conforman un sistema de contratos inteligentes entra en juego para buscar una combinación optima de propiedades, valuando aspectos contrapuestos como la rigidez versus la posibilidad de actualización, una arquitectura monolítica versus una modular o la duplicación de componentes versus la reutilización. (ConsenSys , 2019)

En paralelo es posible y sumamente recomendado acompañar el proceso por alguno de los ciclos de vida de pruebas para Contratos Inteligentes bien estructurados y documentados como el propuesto por Ashray Kakadiya (2017) o Infosys (2018), valiéndose siempre que se posible de herramientas disponibles que permiten la automatización de estos procesos.

En lo relativo a la seguridad durante las actividades de diseño, implementación y verificación de los *Smart Contracts* comprendidas en esta etapa, surgen recomendaciones como formalizar los requerimientos de diseño, estandarizándolo y documentándolo con la rigurosidad que se considere pertinente por las características del proyecto; analizar los posibles ataques acudiendo a documentación de ataques conocidos que mantienen las comunidades de las plataformas como por ejemplo Ethereum; realizar análisis estáticos sobre el diseño y el código en búsqueda de posibles escenarios de ataque y preparar medidas en caso de ocurrencia, utilizar herramientas de testing y desarrollo comprobadas por la comunidad; probar los contratos a fondo en un entorno de pruebas separado y

agregar test cada vez que se descubran nuevos vectores de ataque; utilizar código ya escrito siempre que sea posible.

Mantenerse actualizado con la comunidad de desarrolladores de una plataforma es otra practica esencial que permitirá conocer cualquier nuevo error tan pronto como se descubra y revisar los contratos con tal fin, actualizar a la última versión de cualquier herramienta o biblioteca y adoptar nuevas técnicas de seguridad propuestas que parezcan útiles para el proyecto. (ConsenSys , 2019).

Desarrollo aplicación cliente: Esta etapa involucra las actividades necesarias para el desarrollo de la aplicación cliente que posteriormente hará uso de los servicios implementados en la fase de desarrollo *Blockchain*-SC.

Como los puntos más críticos de la lógica de un sistema orientado a la tecnología *Blockchain* son abordados por separado, un enfoque basado en el modelo de desarrollo por prototipos resulta ser lo más adecuado teniendo en consideración que el resultado de esta etapa es un componente del sistema donde el énfasis este puesto en su interfaz y experiencia de usuario.

Esta etapa inicia con el modelado, refinamiento y priorización de los requisitos establecidos durante la etapa de especificación, se seleccionan y diseñan velozmente con la granularidad que se considere pertinente aquellos a ser abordados durante la iteración correspondiente; seguido, se realiza un prototipo que represente una vista preliminar de una parte del software para ser validado por los interesados. Dependiendo del resultado de la validación, si esta es negativa, se vuelve a modelar el requerimiento, se ajusta el

diseño y se refina el prototipo para una nueva validación, este ciclo se realiza de manera iterativa hasta que los interesados estén de acuerdo con esta primera aproximación. Una vez aceptado el prototipo se procede a la construcción y prueba de la pieza de software real que se constituye como un incremento de la aplicación cliente, para luego abordar un nuevo requisito y realizar el mismo proceso de manera cíclica hasta que la totalidad de los requisitos hayan sido contemplados.

Este enfoque durante la construcción de la aplicación cliente asegura que el software sea de mejor calidad al ser constante el proceso de validación y verificación, al mismo tiempo que el costo en el desarrollo se reduce ya que solo se procede al desarrollo una vez que los prototipos rápidos y económicos de realizar son aprobados.

Integración y despliegue: Durante esta actividad estructural del proceso se integran los desarrollos *Blockchain* y la aplicación cliente, para luego ser desplegados en una red de pruebas que permita probar la interacción entre ambas partes validando y verificando el sistema como un todo para su posterior puesta en producción sobre la plataforma *Blockcahin* principal.

El planteo de la etapa de integración y despliegue surge de la necesidad de llevar adelante determinadas actividades que permitan conectar los dos componentes abordados por separado en este modelo. Como se expuso durante el apartado referido a las DApps, el puente entre la Cadena de Bloques y la aplicación cliente es una API, la cual puede ser desarrollada a mediada como se mencionó con anterioridad, o en el más común de los casos, hacer uso de una existente como por ejemplo Web3. Dependiendo los lenguajes y

tecnologías utilizadas en el desarrollo llevado adelante en ambos componentes, será necesario configurar el consumo de estas interfases por parte de la aplicación cliente.

Existen principios generales al hablar de la conectividad entre una aplicación web con la Cadena de Bloques. Por un lado, es necesario especificar en la configuración de la API el proveedor que permitirá a los usuarios interactuar con los contratos inteligentes desde una interfaz de la app cliente. Sin este proveedor, cada usuario de la aplicación debería descargar la totalidad de la *Blockchain* y funcionar como un nodo independiente, lo cual no sería apropiado ni viable para un usuario regular. Existen alternativas referidas a la instanciación del proveedor dependiendo de la plataforma seleccionada y será necesario realizar actividades de codificación para lograr dicho objetivo. Por otro lado, será necesario realizar la integración del contrato inteligente en la aplicación cliente, por lo general a través de la definición de una ABI (Interfaz Binaria de aplicación).

Una vez integradas ambas partes es necesario desplegarlas en una red de pruebas, por lo general las plataformas más usadas cuentan con una llamada *testnet*, así como también con la posibilidad de virtualizar una red de manera local. Este paso permitirá realizar las pruebas de sistema y capturar posibles errores o escenarios de falla y tomar medidas previo a su puesta en producción.

Como puede notarse la integración y despliegue demanda de prácticas y recaudos particulares, motivo por el cual se justifica definirla como una actividad estructural independiente en MDSOB.

Conclusiones y trabajos futuros

A raíz de lo desarrollado a lo largo del trabajo, es posible señalar que la tecnología *Blockchain* ha comenzado a posicionarse recientemente en el plano productivo y empresarial donde los errores cometidos durante el ciclo de vida de desarrollo se vuelven costosos y en muchos casos críticos en el contexto del sistema. Esta situación desencadena la necesidad de colocar bajo la lupa de la Ingeniería en Software a dicha tecnología con el fin de analizar desde su experticia el abordaje en muchos casos deficiente de los proyectos que involucran a *Blockchain* como uno de sus principales componentes.

La similitud entre lo que fue la llamada crisis del software, y la situación actual que envuelve a la tecnología *Blockchain*, revive la idea de antaño de una crisis crónica, donde la aparición de nuevos paradigmas tecnológicos para la construcción de software remueve los cimientos de la disciplina ingenieril, demandando la adaptación e inclusión de nuevas herramientas, métodos y procesos que promuevan el software de calidad (mantenible, eficiente, seguro, confiable y aceptable). En relación a lo antes expuesto, la percepción y evaluación de las prácticas y enfoques disimiles de los participantes de la encuesta dentro de la comunidad de desarrolladores analizada, reafirma el supuesto acerca de la ausencia de un abordaje normalizado que haga uso efectivo de los principios de la Ingeniería en Software para el desarrollo *Blockchain.*

Resulta clave, aunque difícil de conseguir, acompañar de manera sincrónica la evolución de esta tecnología con buenas prácticas para su desarrollo e implementación.

Partiendo del estudio realizado, esta situación parece ser utópica. La falta de buenas prácticas tiende a responder principalmente a dos causas, por un lado, a la carente adaptación a este nuevo paradigma tecnológico de los métodos y herramientas existentes, y por él otro, a la tendencia arraigada en la mayoría de los equipos de desarrollo y empresas de operar con agilidad.

Teniendo en cuenta el rol fundamental que desempeña el software para el progreso económico y como ventaja competitiva de las empresas, no resulta aventurado confirmar la aseveración expuesta con anterioridad. Es evidente que la prioridad actualmente se encuentra en el desarrollo rápido de soluciones. Si bien es indudable que el enfoque adaptativo en la forma en que se encaran proyectos de software, bajo tecnologías bien conocidas, reporta grandes beneficios y ventajas sobre los métodos más planificados y predictivos, no es así en el caso de la tecnología *Blockchain*, donde su inmadurez, complejidad tecnológica y múltiples alternativas de implementación requieren de un análisis minucioso y un profundo entendimiento previo y durante el desarrollo.

Por otro lado, es posible notar nuevos desafíos al momento de pensar en *Blockchain* dentro de un ambiente productivo y empresarial. Para que se implemente y capture el valor dentro de este entorno, *Blockchain* debe poder conectarse a sistemas, operaciones y comportamientos heredados. Este es un problema muy real que a menudo plantea altas barreras operativas y tecnológicas, especialmente porque las ganancias de la resistencia a ciberataques de *Blockchain* se pueden negar conectándose a sistemas heredados vulnerables. Esta situación lleva a prestar especial atención al desarrollo de

interfases de comunicación que no resulten en el deterioro de los beneficios de seguridad intrínsecos de la tecnología *Blockchain* al incluir potenciales vectores de ataque originados por los sistemas con los que interactúa.

Es importante remarcar también que la investigación sobre los avances en el campo del desarrollo de software orientado a *Blockchain* permitió recapitular valiosas prácticas, herramientas y principios que pueden ser aplicados a lo largo del ciclo de vida de desarrollo, empero, más allá de estas propuestas, es notoria la ausencia de modelos, y lineamientos generales e independientes de las plataformas, que articulen un proceso ordenado que conduzca a mejores resultados en el software producido.

Lo expuesto con anterioridad genera el vacío que otorga valor a la propuesta de este trabajo. El modelo de desarrollo MDSOB retoma conceptos y principios de la ingeniería de software tradicionales, los combina con las particularidades tecnológicas de *Blockchain* y da lugar a la aplicación de los avances introducidos en el campo. El MDSOB permite ordenar y guiar el desarrollo de procesos sistemáticos hacia un enfoque ingenieril que posibilite arribar a un producto de software de mayor calidad mientras mantiene el grado de abstracción necesario para hacer frente a las varianzas propias de la falta de estandarización en torno a *Blockchain* y los *Smart Contracts*. Por otra parte, MDSOB posibilita combinar resultados agiles con la planificación necesaria para una adopción más correcta de la tecnología *Blockchain* que mitigue posibles riesgos durante el ciclo de vida de desarrollo.

Los resultados de este Trabajo Final de Grado sirven como aporte para comprender las particularidades de la tecnología *Blockchain,* presenta evidencia de la

abierta comprensión y formas de implementación como puntapié inicial para detallar procesos específicos en base a una plataforma y proyecto determinado.

En lo particular, se pretende retomar el modelo MDSOB y evaluar su efectividad y plasticidad mediante su utilización en la definición de un proceso aplicable a un proyecto real basado en la plataforma Ethereum, siendo que en la actualidad la misma se reconoce como la más afianzada y madura dentro del ecosistema de las aplicaciones distribuidas en *Blockchain*.

Bibliografía

Almeida, S., Albuquerque, A., & Silva, A. (2018). An Approach to Develop Software that Uses Blockchain. *Software Engineering and Algorithms in Intelligent Systems, 763*, 346-355. Springer International Publishing AG.

Arias Odón, F. G. (2012). *El Proyecto de Investigacion.* Editorial Episteme.

Bartoletti, M., & Pompianu, L. (Marzo de 2017). Lecture Notes in Computer Science. *An Empirical Analysis of Smart Contracts: Platforms, Applications, and Design Patterns*. doi:DOI: 10.1007/978-3-319-70278-0_31

Bashir, I. (2018). *Mastering Blockchain* (Segunda ed.). Birmingham, Reino Unido: Packt Publishing.

Bit2Me. (s.f.). *Qué es un bloque dentro de la blockchain*. Obtenido de https://academy.bit2me.com/que-es-un-bloque-dentro-de-la-blockchain/

Brooks, F. (Abril de 1987). No Silver Bullet Essence and Accidents of Software Engineering. *Computer, 20*(4), 10-19. doi:10.1109/MC.1987.1663532

Chakraborty, P., Shahriyar, R., Iqbal, A., & Bosu, A. (11-12 de Octubre de 2018). Understanding the Software Development Practices of Blockchain Projects: A Survey. *12th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement(Articulo No. 28)*. Oulu, Finlandia. doi:10.1145/3239235.3240298

ConsenSys . (2019). *Ethereum Smart Contract Best Practices* . Obtenido de https://consensys.github.io/smart-contract-best-practices/

De La Cruz Casaño, C. (Enero-Junio de 2016). Metodología de la investigación tecnológica en ingeniería. *Revista Ingenium, 1*(1), 43-46. doi:http://dx.doi.org/10.18259/ing.2016007

Diar Ltd . (01 de Octubre de 2018). Venture Capital Firms Go Deep and Wide with Blockchain Investments. *The Digital Assets & Regulation Trade Publication, 2*(39). Obtenido de https://diar.co/volume-2-issue-39/#2

Diffie, W., & Hellman, M. (Noviembre de 1976). New Directions in Cryptography. *Journal IEEE Transactions on Information Theory, 22*(6), 29-40. doi:10.1109/TIT.1976.1055638

Disparte, D. (20 de Mayo de 2019). *Why Enterprise Blockchain Projects Fail*. Obtenido de Forbes: https://www.forbes.com/sites/dantedisparte/2019/05/20/why-enterprise-blockchain-projects-fail/#5cdd46594b96

Fridgen, G., Lockl, J., Radszuwill, S., Rieger, A., Schweizer, A., & Urbach, N. (Agosto de 2018). A Solution in Search of a Problem: A Method for the Development of Blockchain Use. *24th Americas Conf. Information Sysems*. Nueva Orleans, Los Angeles, USA.

Garousi, V., Coşkunçay, A., Betin-Can, A., & Demirörs, O. (2014). A Survey of Software Engineering Practices in Turkey. *Journal of Systems and Software*(108), 148–177. doi:10.1016/j.jss.2015.06.036

Gibbs, W. W. (Septiembre de 1994). Software's Chronic Crisis. *Scientific American, 271*(3), 86-95.

Haber, S., & Stornetta, S. W. (1991). How to Time-Stamp a Digital Document. *Journal of Cryptology, 3*(2), 99-111. doi:https://doi.org/10.1007/BF00196791

Infosys. (2018). Assuring success in blockchain implementations by engineering quality in validation. Gandhinagar, Gujart, India.

Kakadiya, A. (Diciembre de 2017). Block-Chain Oriented Software testing approach. *International Research Journal of Engineering and Technology (IRJET), 4*(12), 1593-1597.

Lamport, L. (Mayo de 1998). The Part-Time Parliament. *ACM Transactions on Computer Systems, 16*(2), 133-169.

Marchesi, M., Marchesi, L., & Tonelli, R. (Rusia de Octubre de 2018). Software Engineering Conference Russia . *An Agile Software Engineering Method to Design Blockchain Applications*. Moscow.

Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). *Handbook of applied cryptography* (Boca Raton, xiii, 780, 1997. ed.). CRC Press.

Nakamoto, S. (1 de Noviembre de 2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* Obtenido de http://bitcoin.org/bitcoin.pdf

Orcutt, M. (5 de Abril de 2019). *El masivo historial de robos demuestra que 'blockchain' no es inhackeable*. Obtenido de MIT Technology Review: https://www.technologyreview.es/s/10958/el-masivo-historial-de-robos-demuestra-que-blockchain-no-es-inhackeable

Porru, S., Pinna, A., Michele, M., & Tonelli, R. (Febrero de 2017). Blockchain-oriented Software Engineering: Challenges and New Directions.

Pressman, R. (2010). *Ingeniería del software. Un enfoque práctico.* Mexico: McGRAW-HILL INTERAMERICANA EDITORES.

PwC. (2018). *PwC 's 2018 Global Blockchain Survey.* Hong Kong. Obtenido de https://www.pwccn.com/global-blockchain-survey-2018

Sommerville, I. (2011). *Ingeniería de Software.* México: PEARSON EDUCACIÓN.

Swan, M. (2015). *Blockchain-Blueprint for a New Economy.* United States of America: O'Reilly Media.

Szabo, N. (1994). *Smart Contracts [Contratos inteligentes].* Obtenido de http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

Tonelli, R., Destefanis, G., Marchesi, M., & Ortu , M. (7 de Febrero de 2018). Smart Contracts Software Metrics: a First Study. doi:10.13140/RG.2.2.25506.12483

Wessling, F., Ehmke, C., Hesenius, M., & Gruhn, V. (Mayo de 2018). How Much Blockchain Do You Need? Towards a Concept for Building Hybrid DApp Architectures. *2018 ACM/IEEE 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*. Gothenburg, Suiza. doi:https://doi.org/10.1145/3194113.3194121

Wöhrer, M., & Zdun, U. (2018). Design Patterns for Smart Contracts in the Ethereum Ecosystem. *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. Halifax, NS, Canada. doi:10.1109/Cybermatics_2018.2018.00255

Wöhrer, M., & Zdun, U. (Marzo de 2018). Smart contracts: security patterns in the ethereum ecosystem and solidity. *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, 2-8. Campobasso. doi:10.1109/IWBOSE.2018.8327565

Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., . . . Rimba, P. (Abril de 2017). A Taxonomy of Blockchain-Based Systems for Architecture Design. Gothenburg, Suecia.

Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2018). *Blockchain Technology Overview.* National Institute of Standards and Technology Internal Report 8202, National Institute of Standards and Technology, Estados Unidos. Obtenido de https://doi.org/10.6028/NIST.IR.8202

Anexos

Anexo A – Formulario descriptivo del trabajo final de graduación.

AUTORIZACIÓN PARA PUBLICAR Y DIFUNDIR TESIS DE POSGRADO O GRADO A LA UNIVERIDAD SIGLO 21

Por la presente, autorizo a la Universidad Siglo21 a difundir en su página web o bien a través de su campus virtual mi trabajo de Tesis según los datos que detallo a continuación, a los fines que la misma pueda ser leída por los visitantes de dicha página web y/o el cuerpo docente y/o alumnos de la Institución:

Autor-tesista *(apellido/s y nombre/s completos)*	Zamora Fernando Gabriel
DNI *(del autor-tesista)*	37729039
Título y subtítulo *(completos de la Tesis)*	**Hacia una Ingeniería en Software orientada a *Blockchain*.** *Propuesta de un modelo de desarrollo de software basado en el análisis de la tecnología y las prácticas de desarrolladores Blockchain en Argentina.*
Correo electrónico *(del autor-tesista)*	fernando-zamora-@hotmail.com
Unidad Académica *(donde se presentó la obra)*	Universidad Siglo 21

Otorgo expreso consentimiento para que la copia electrónica de mi Tesis sea publicada en la página web y/o el campus virtual de la Universidad Siglo 21 según el siguiente detalle:

Texto completo de la Tesis *(Marcar SI/NO)*[1]	SI
Publicación parcial *(Informar que capítulos se publicarán)*	

Otorgo expreso consentimiento para que la versión electrónica de este libro sea publicada en la página web y/o el campus virtual de la Universidad Siglo 21.

Lugar y fecha: _____

Firma autor-tesista

Fernando Gabriel Zamora

Aclaración autor-tesista

Esta Secretaría/Departamento de Grado/Posgrado de la Unidad Académica:

_____certifica

que la tesis adjunta es la aprobada y registrada en esta dependencia.

Firma Autoridad

Aclaración Autoridad

Sello de la Secretaría/Departamento de Posgrado

[1] Advertencia: Se informa al autor/tesista que es conveniente publicar en la Biblioteca Digital las obras intelectuales editadas e inscriptas en el INPI para asegurar la plena protección de sus derechos intelectuales (Ley 11.723) y propiedad industrial (Ley 22.362 y Dec. 6673/63. Se recomienda la NO publicación de aquellas tesis que desarrollan un invento patentable, modelo de utilidad y diseño industrial que no ha sido registrado en el INPI, a los fines de preservar la novedad de la creación.