BlockGuitars: una herramienta para las transacciones de instrumentos musicales mediante blockchain

Damian Castelli Lluch[1] and Anibal Tallarico[1] and Claudia Pons[1,2,3]

[1]Universidad Nacional de La Plata, Facultad de Informática, LIFIA. Buenos Aires, Argentina
[2]Universidad Abierta Interamericana, Facultad de Tecnología Informática, CAETI
[3] Comisión de Invetigaciones Científicas de la Provincia de Bs. As.
claudia.pons@uai.edu.ar

Abstract. Blockchain y smart contracts son novedosas tecnologías que permiten registrar de forma eficaz y segura transacciones de elementos virtuales, como criptomonedas. Pero también se pueden aplicar a otros ámbitos. En este artículo se describe la construcción de un sistema de registro de instrumentos musicales aplicando estas nuevas tecnologías. El aporte, en primer lugar, es poner a disposición de los músicos una herramienta segura y distribuida que les ofrecerá una alternativa para enfrentar los frecuentes hurtos de estos elementos. Por otra parte, la descripción de la metodología seguida para el desarrollo del sistema, así como las decisiones de diseño adoptadas, la base tecnológica seleccionada y el análisis legal realizado, son recursos de utilidad para los desarrolladores que inicien la construcción de un sistema similar en otro dominio.

Keywords: Blockchain, Ethereum, Smart Contract, Solidity, Instrumentos musicales.

1 Introducción

1.1 La problemática del registro de instrumentos musicales en Argentina

En Argentina la Cámara de Instrumentos Musicales, Audio, Video e Iluminación (CAIMAVI) agrupa a cuarenta representantes o distribuidores y la Cámara Argentina de Fabricantes de Instrumentos Musicales (CAFIM) reúne a 200 socios, desde luthiers a pequeñas empresas, ambos, concentran el 85 % del mercado. Sin embargo estos organismos no ofrecen una solución para registrar la adquisición o posesión de instrumentos musicales y la informalidad del sector facilita la circulación de instrumentos robados.

Los músicos que han sufrido robos de instrumentos utilizan las redes sociales para buscar sus objetos sustraídos, ya sea compartiendo una publicación, o esperando a que el instrumento sea ofrecido en las redes. En los casos de instrumentos seriados, se tiene la ventaja de que el instrumento puede identificarse, en los casos de los instrumentos no seriados (Luthiers), la situación es aún peor. En los pocos casos que se recupera, las víctimas negocian por su cuenta sin intervención del estado. Las denuncias a la policía en muchos casos no son realizadas, ya sea porque no se posee factura

de compra (en los casos de instrumentos de segunda mano), como por la poca credibilidad tanto de la policía como del sistema jurídico. Sindicatos, agrupaciones y músicos lanzan campañas para desestimar las compras de instrumentos sustraídos [1].

Existen algunas empresas y bancos que aseguran instrumentos ante hurtos. Sin embargo el costo del servicio es desproporcionadamente alto. Por otra parte, existen sitios en línea para denunciar / consultar instrumentos robados como StolenGuitarRegistry [2] o PlayChecked [3], servicio para el registro de instrumentos. Ambos servicios son gratuitos y ofrecen registro de instrumentos a través de su número de serie. Estos proyectos carecen de un marco legal y sólo constituyen un sistema cerrado para el músico ofreciéndole un servicio básico. Paralelamente existen servicios como ScreamingStone [4], quienes utilizando técnicas de, SEO (Search Engine Optimization) ofrecen el servicio de recuperación de instrumentos robados.

1.2 Blockchain como mecanismo de registro seguro

Para cualquier tipo de propiedad de valor es importante mantener registros precisos que permitan a los propietarios identificarse como tales, de manera que puedan ser utilizados para proteger sus derechos, resolver disputas, asegurar que las transferencias de propiedad son realizadas y prevenir fraudes. En general se requiere de una entidad central reguladora, tal como fabricantes o agencias gubernamentales, sobre quien se deposita la confianza para el mantenimiento de estos registros.

Sin embargo, a partir del surgimiento de Blockchain [5] comenzó a vislumbrase un paradigma diferente, que permite pensar en redes donde los participantes intercambian algo que todos valoran, pero ofreciendo la ventaja de no requerir de una entidad central reguladora de las transacciones, porque la implementación misma basada en consenso se asegura de la validez y deja, por ende, sin sentido a la entidad centralizada. El mejor ejemplo de uso de Blockchain es Bitcoin [6], una critpomoneda desregulada con más de 10 años de circulación en el mercado.

Técnicamente hablando Blockchain es una tecnología DLT (del inglés Distributed Ledger Technology) que nos provee la facilidad de acceder a un repositorio compartido por todos los miembros de una red que, con mecanismos de acceso seguro, pueden ver la información de todo lo que se escribe en ese libro. Este repositorio compartido emula el concepto del libro contable de una entidad (en inglés: ledger) que representa sus activos y movimientos y lo traslada a una infraestructura de tecnología en la nube que permite que la misma funcionalidad se represente en forma segura y dando acceso a todos los participantes de la red.

Blockchain en ingles significa “cadena de bloques” porque es una estructura de datos en la que la información contenida se agrupa en conjuntos (bloques) a los que se les añade meta información relativa a otro bloque anterior en una línea temporal, de manera que gracias a técnicas criptográficas la información contenida en un bloque sólo puede ser repudiada o editada modificando todos los bloques posteriores. Esta propiedad permite su aplicación en entorno distribuido de manera que la estructura de datos blockchain puede ejercer de base de datos pública no relacional que contenga un histórico irrefutable de información. [7]. Cualquier modificación malintencionada

debería hacerse de forma coordinada en todas las copias que existan, lo que lo hace virtualmente imposible.

1.3 Contratos inteligentes

Otro concepto fundamental en esta área son los contratos inteligentes (en inglés Smart Contracts o SC) [8], que son programas informáticos que facilitan, aseguran, hacen cumplir y ejecutan acuerdos registrados entre dos o más partes. El concepto fue desarrollado en el contexto de la criptomoneda Ethereum [9], y permite agregar código inteligente en forma de reglas de negocio "inmutables" que serán ejecutadas con cada transacción según se programen.

Un contrato especifica un acuerdo entre partes con ciertas acciones y condiciones de cumplimiento. Un contrato se hace "inteligente" cuando incluye código y lógica que ejecuta las acciones y fuerza la aplicación de las condiciones, es decir, código que hace que se cumpla el contrato sin necesidad de que ninguna de las partes intervenga manualmente. El Smart Contract cumple una interfaz estándar, se desarrolla en un lenguaje concreto, generalmente Solidity [10], y se compila y ejecuta dentro de la red, en la Ethereum Virtual Machine (EVM).

1.4 Una solución tecnológica para registrar instrumentos musicales

Blockchain y los smart contracts han demostrado su efectividad durante estos años en un entorno tan sensible como es el económico. Más allá de eso, esta novedosa tecnología que permite registrar de forma eficaz y segura transacciones de elementos virtuales también se puede aplicar a otros ámbitos aparte del económico o las criptomonedas. Por ejemplo el arte digital, derechos de autor, o la votación electrónica.

En este artículo se describe su uso para construir un sistema de registro de instrumentos musicales que ofrece mayor seguridad y ventajas que los sistemas actuales y que intenta ofrecer una solución a los problemas descriptos en la sección 1.1.

El artículo está organizado de la siguiente forma. La sección 2 describe la funcionalidad de la aplicación y los pasos seguidos para su desarrollo. La sección 3 discute los desafíos legales que enfrentan este tipo de aplicaciones. Finalmente la sección 4 presenta conclusiones y líneas de trabajo futuras.

2 Registro de transacciones de instrumentos musicales mediante blockchain

En esta sección se describe la funcionalidad del sistema de registro de instrumentos musicales que se ha desarrollado y los pasos seguidos para su implementación.

2.1 Funcionalidad del sistema BlockGuitars

El sistema de registro de instrumentos que se ha desarrollado se denomina BlockGuitars y ofrece la siguiente funcionalidad:

- Mantiene una copia propia de la cadena de bloques de blockchain, que será verificada y se utilizará para extraer los datos relativos a las propiedades.
- Detecta y procesa las transacciones para extraer la información relativa a las propiedades que representan.
- Ofrece a los propietarios un sistema de acceso a la información extraída, de modo que puedan consultar y realizar operaciones sobre sus registros.
- Ofrece a terceros la posibilidad de consultar la información extraída.
- Mantiene la información actualizada, de manera que las nuevas emisiones y transferencias de propiedades que sean realizadas sean identificadas por el sistema tras la validación de las transacciones recibidas.

2.2 Proceso de desarrollo del sistema

El desarrollo de aplicaciones basadas en blockchain es una disciplina relativamente nueva, por lo que todavía no existen procesos estandarizados dentro de la ingeniería de software, pero si existen algunas propuestas aceptadas por la comunidad. Para el desarrollo de este proyecto se utilizó la propuesta presentada en [11]. Este proceso consta de 8 etapas las cuales se describen a continuación:

Etapa 1 - Definir el objetivo del sistema.
Como se dijo anteriormente el objetivo del sistema es registrar, de manera segura y sin intermediarios, las transacciones de instrumentos musicales tales como alta, baja y transferencias de estos entre personas.

Etapa 2 – Identificar los actores (humanos y sistemas externos)
Los actores identificados tal como se muestra en el diagrama de casos de uso de la fig.1, son:

- *Administrador*: Es el usuario que crea el sistema.
- *Persona*: Este usuario puede dar de alta el instrumento, darlo de baja, listar los instrumentos y transferirlos a otras personas.

Etapa 3 - Definir historias de usuario
En esta etapa se redactaron los requerimientos del sistema en base a historias de usuario. Se muestra aquí un resumen de las historias más relevantes:

- Creación del sistema: El administrador crea el sistema.
- Listado de instrumentos: El usuario visualiza los instrumentos que se encuentran registrados en el sistema.
- Alta de instrumento: El usuario da de alta un instrumento ingresando marca y código de serie.
- Baja de instrumento: El usuario selecciona un instrumento de su propiedad y lo da de baja.
- Transferencia de instrumento: El usuario selecciona un instrumento de su propiedad y lo transfiere a la dirección que desee.

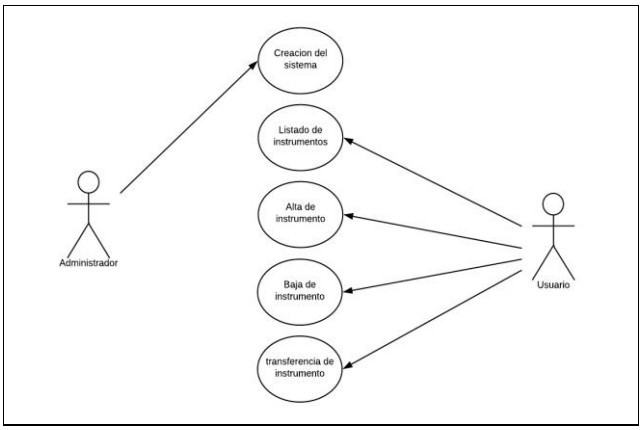

Fig. 1. - Diagrama de casos de uso.

Etapa 4 – Definir la arquitectura del sistema

El sistema se dividió en dos subsistemas separados:

- El sistema blockchain compuesto por smart contracts, este interactúa con el exterior exclusivamente a través de transacciones blockchain, sus actores son reconocidos a través de la dirección ethereum; puede utilizar librerías y contratos externos y puede generar transacciones a otros contratos. Los contratos están escrito en Solidity.
- El sistema externo que interactúa con el anterior enviando transacciones blockchain y recibiendo resultados a través de interfaces. La interfaz web es independiente y se ha desarrollado mediante tecnologías web del lado del cliente.

La conexión entre ambas partes se realiza mediante el uso de una librería proporcionada por la propia plataforma de Ethereum.

Los Smart Contracts fueron modelados mediante diagramas de clase UML con estereotipos (ver Fig. 2). Al igual que una clase, un Smart Contract tiene una estructura de datos, funciones públicas y privadas, y pueden heredar de uno o más SCs. Un SC puede enviar mensajes a otros SCs, que residen en el mismo blockchain.

Otros conceptos específicos de Ethereum SC son los eventos, que sirven para notificar al mundo exterior sobre la ocurrencia de algo relevante.

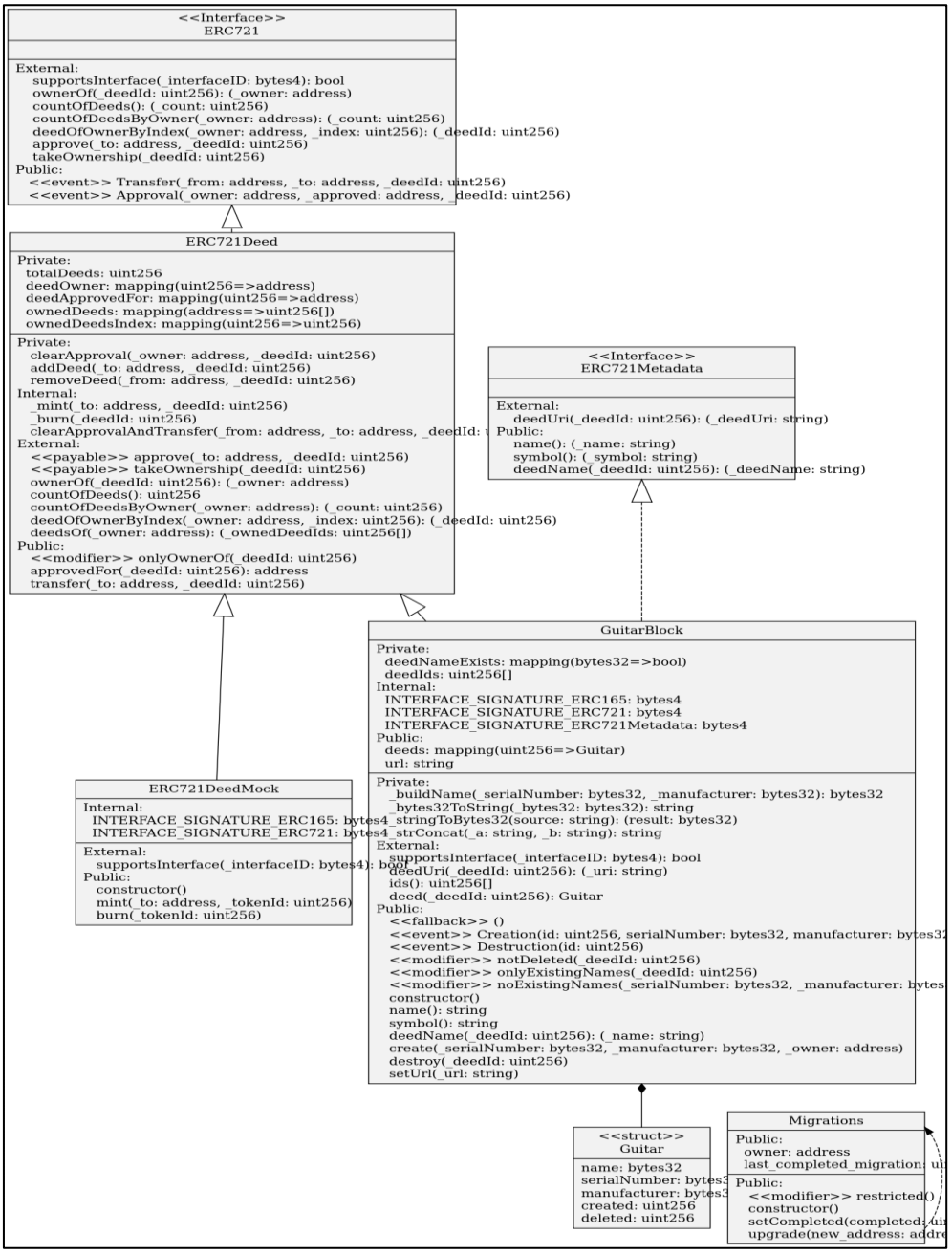

Fig. 2. - Diagrama de clases de los Smart Contracts

Etapa 5- Diseño del subsistema smart contract

La quinta etapa consistió en el diseño y descomposición de los smart contracts. En la Fig. 2 se muestran los principales contratos. Se diseñó un smart contract que hereda de las interfaces ERC721 y ERC 721 Metadata y del contrato ERC 721 Deed [9]. También se utilizaron contratos de la librería OpenZepellin como Pausable y Safe-Math [9]. Para cada smart contract se definió la estructura y diagrama de estados.

El contrato posee una estructura para almacenar los instrumentos con los siguientes campos: *name, serialNumber, manufacturer, created, deleted*.

Se definieron 2 eventos, uno para crear y otro para destruir un objeto:

-event Creation(uint256 indexed id, bytes32 indexed serialNumber, bytes32 indexed manufacturer, address owner);

-event Destruction(uint256 indexed id);

Se definieron 3 estructuras de datos. Una para enlazar el propietario al instrumento (mapping), una para controlar que no haya elementos repetidos (mapping), y otra para tener todos las escrituras (array):

```
mapping (uint256 => Guitar) public deeds;
mapping (bytes32 => bool) private deedNameExists;
uint256[] private deedIds;
```

Paralelamente se diseñaron los test unitarios para los contratos., utilizando Mocha.js y Chai.js.

Etapa 6: Diseño del subsistema externo

Durante esta etapa se realizaron las siguientes tareas:

- Se redefinieron los actores y las user stories, agregando el nuevo actor (pasivo) representado por los smart contract.
- Se decidió la arquitectura del sub sistema: La arquitectura elegida es una aplicación web responsive a la cual acceden los usuarios para poder realizar las transacciones de los instrumentos.
- Se definió la interfaz de usuario de los módulos relevantes (por ejemplo la interfaz ilustrada en la Fig.3).

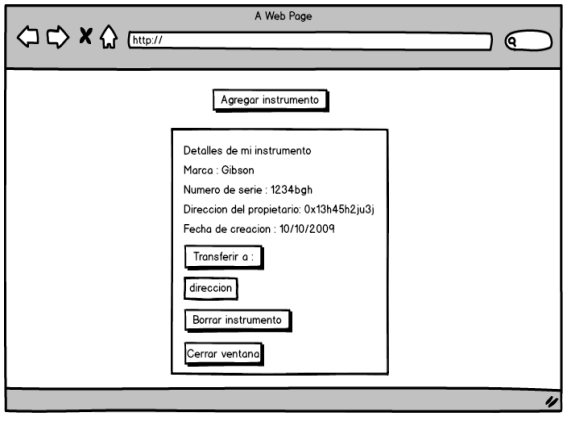

Fig. 3. - Detalle del instrumento propio y las operaciones permitidas

Etapa 7 - Codificar y testear los sistemas en paralelo

Durante esta etapa se codificaron y testearon los smart contracts empezando por sus estructuras de datos y funciones. La figura 4 muestra uno de los Smart contract desarrollado mediante el framework truffle.

```
pragma solidity ^0.4.18;

import "zeppelin/contracts/math/SafeMath.sol";
import "zeppelin/contracts/lifecycle/Pausable.sol";
import "./ERC721Deed.sol";
import "./ERC721Metadata.sol";

/*
  Notes on this ERC721 implementation:
  Notas sobre esta implementación de ERC721:
 */

contract GuitarBlock is ERC721Deed, ERC721Metadata, Pausable {

  using SafeMath for uint256;

  /* Events */
  // When a deed is created by the contract owner.
  // Cuando el propietario del contrato crea una escritura.

  event Creation(uint256 indexed id, bytes32 indexed serialNumber, bytes32 indexed manufacturer, address owner);

  // When a deed needs to be removed. The contract owner needs to own the deed in order to be able to destroy it.
  // Cuando una escritura necesita ser eliminada. El propietario del contrato debe ser el propietario de la escritu

  event Destruction(uint256 indexed id);

  // The data structure of the guitar
  // La estructura de datos de la guitarra.

  struct Guitar {
    bytes32 name;
    bytes32 serialNumber;
    bytes32 manufacturer;
    uint256 created;
    uint256 deleted;
  }
```

Fig. 4. –Smart *contract – parte 1*

Luego se implementaron las user stories del subsistema externo mediante el framework vue.js y la librería web3js que interactúa con el smart contract. La Fig. 5 muestra la implementación de una de las historias.

```
deleteGuitarDeed: function() {
  const destroyDeed = async () => {
    var self = this;
    GuitarBlock.defaults({from: this.userAccount, gas: 900000 });
    let deed = await GuitarBlock.at(this.contractAddress);
    this.processingMessage = "Eliminando Guitarra. Esto puede tardar un rato...";
    this.showSpinner=true;
    this.displayRegistrationComponents = false;
    this.sleep( milliseconds: 1000);
    try {
      let result = await deed.destroy(this.guitarId);
    } catch (error) {
      console.log(error.message);
      this.processingMessage = error.message;
      alert(error.message);
      this.showSpinner=false;
      this.displayRegistrationComponents = true;
      return;
    }
    this.processingMessage = "Tu guitarra ha sido eliminada!";
    this.guitarOwner = this.newOwnerAddress;
    this.showSpinner=false;
    this.displayRegistrationComponents = false;
    this.initAccounts();
    this.loadAllGuitars();
  }

  this.initAccounts();
```

Fig. 5. - Implementación borrado de instrumento

Etapa 8 Integrar, testear y hacer deploy

Para iniciar el sistema deben ejecutarse los siguientes comandos:

npm install: sirve para instalar las librerías necesarias especificadas en el proyecto

truffle compile : Sirve para compilar los archivos fuentes del contracto.

ganache-cli -p 8545: mediante este comando se inicia ***ganache*** en el servidor local en el puerto 8545, esto proveerá una billetera con 10 direcciones y sus claves privadas.

truffle migrate: Se utiliza para ejecutar las migraciones para implementar los contratos. Esto brindara la dirección del contrato, la que luego se agrega en la variable *contractAddress* perteneciente al archivo guitarappvue.js del sistema para poder trabajar con el contrato creado.

npm run build :

npm run server:

En el inicio de la aplicación se podrá visualizar el listado de los instrumentos registrados, con los datos de código de serie y marca como así también un buscador y el enlace al alta de instrumento.

2.3 Stack Tecnológico

En esta sección se describen las principales tecnología que se utilizaron para el desarrollo del sistema, con el objetivo de compartir la experiencia y asi facilitar la tarea a los desarrolladores que deban encarar la construcción de sistemas de similares características.

La aplicación final está dividida en un contrato y una interfaz web. La interfaz web es independiente y se ha desarrollado mediante tecnologías web del lado del cliente. El contrato está escrito en Solidity. La conexión entre ambas partes se realiza mediante el uso de una librería proporcionada por la propia plataforma de Ethereum.

Las principales tecnologías usadas fueron:

- Ethereum: se usó la red de Ethereum para publicar contratos.
- Plugin Metamask para navegadores Google Chrome/Mozilla Firefox: permitió conectar el navegador a la red Ethereum en sencillos pasos.
- Remix IDE: se trata de un entorno de programación para Solidity totalmente online. Permitió programar y ejecutar el Contrato Inteligente, realizar llamados a las funciones internas del contrato, añadir y modificar información del mismo.
- Red privada Kovan para solicitar Kovan Ether (KETH). Publicar contratos en la red Ethereum conlleva un gasto en ETHER. Existen redes de prueba dentro de la red Ethereum como Kovan que permiten ejecutar contratos inteligentes sin gastar ETHER real.
- Saldo Kovan Ethereum de prueba: A través del siguiente sitio se obtuvo saldo para realizar pruebas https://faucet.kovan.network/
- Tecnologías Web: HTML5, Javascript nativo, Javascript Frameworks (Node JS a través de NVM, Jquery, Web3, NPM), CSS Frameworks (Bootstrap) y Solidity. https://github.com/nvm-sh/nvm , https://nodejs.org/, https://www.npmjs.com/
- Web3.js: Es una API en Javascript compatible con Ethereum que implementa las especificaciones genéricas RPC en formato JSON. Para que la aplicación funcione

en Ethereum se puede usar el objeto web3 proporcionado por la biblioteca web3.js. La librería por dentro se comunica con un nodo local a través de llamadas RPC. Web3.js funciona con cualquier nodo Ethereum que expone una capa RPC. Web3 contiene el objeto eth (web3.eth) para interacciones específicas de cadena de bloques de Ethereum y el objeto shh (web3.shh) para la interacción de Whisper. https://github.com/ethereum/web3.js/

- Truffle: https://www.trufflesuite.com/es el framework más popular para el desarrollo en Ethereum actualmente, posee las siguientes características:
    - Compilación, enlace y despliegue de Smart contracts desde el propio framework
    - Depuración y testing automatizado de contratos. Truffle utiliza el framework de prueba Mocha.js y Chai.js para las afirmaciones proporcionando un marco sólido desde el cual poder escribir pruebas.
    - Framework con scripts de despliegue y migraciones en redes públicas y privadas
    - Acceso a cientos de paquetes externos y gestión con EthPM & NPM
    - Consola interactiva para comunicación directa con los contratos
    - Interacción con contratos mediante scripts externos.
- Mocha: Mocha es un framework de pruebas de JavaScript que se ejecuta en Node.js. Brinda la posibilidad de crear tanto tests síncronos como asíncronos de una forma muy sencilla. https://mochajs.org/
- Chai: Es un librería de aserciones, la cual se puede emparejar con cualquier framework de pruebas de Javascript. Chai tiene varias interfaces: assert, expect y should, que permiten al desarrollador elegir el estilo que le resulte más legible y cómodo a la hora de desarrollar sus tests. https://www.chaijs.com/
- Openzeppelin: Es una biblioteca para el desarrollo seguro de contratos inteligentes. Proporciona implementaciones de estándares como ERC20 y ERC721. https://openzeppelin.org/
- EthPM: Gestor de paquetes inmutable para consumir, distribuir y administrar de manera fácil y segura cualquier sistema de contrato inteligente. https://www.ethpm.com/
- ERC721ExampleDeed: Biblioteca basada en OpenZeppelin para la escritura de objetos como propiedad. https://github.com/nastassiasachs/ERC721ExampleDeed

3 El contexto legal de BlockGuitars y trabajos relacionados

Uno de los principales retos de BlockGuitars es la dificultad de unir tres mundos, el tecnológico, el musical y el legal. Los contratos inteligentes escritos por los técnicos y los contratos propiamente dichos, escritos por los especialistas en leyes deben integrarse. El reto no es sólo trasladar todo el lenguaje legal a un mundo computacional, sino que además se tienen que dar algunos avances para lograr su validez jurídica y estandarización en la industria. Aun así, el concepto de contratos inteligentes avanza rápidamente y tiene el potencial de causar disrupciones significativas en las industrias

creativas, especialmente en el contexto de la gestión de derechos y los sistemas de pago de regalías.

Sin embargo, aún resulta incierto el resultado de unir el mundo tecnológico con el legal. Parte importante de la incertidumbre está vinculada al marco jurídico que resta establecer para regular la entidad legal de una tecnología global como blockchain, donde los diferentes países podrían adoptar posturas diferentes.

3.1 La validez jurídica de BlockGuitars respecto a la posesión del bien

El Código Civil y Comercial de la Nación, en su artículo 1909 - Posesión- establece que "Hay posesión cuando una persona, por sí o por medio de otra, ejerce un poder de hecho sobre una cosa, comportándose como titular de un derecho real, lo sea o no".

Si bien un instrumento musical no es un bien registrable como una casa o un vehículo, el comprador de un instrumento musical robado registrado en BlockGuitars no podría alegrar que ignoraba que compró un bien robado, por lo cual no podría hacer valer el derecho por encima del dueño desapoderado.

Un sistema como BlockGuitars al permitir establecer el historial de propiedad de un bien evitaría la compra-venta de instrumentos robados, ya que habría una interrupción en la cadena de tenedores que no podría ser desconocida por el poseedor actual y por lo tanto no se podría sostener que el instrumento se compró de buena fe a una persona que no es la que figura como propietario del bien.

3.2 La validez jurídica de BlockGuitars en la herencia y el testamento

A la hora de distribuir la herencia, un sistema de blockchain no podría ser viable en términos legales ya que Argentina no admite el testamento en su forma digital. La situación es diferente en otros países, como España. El testamento digital aparece regulado por primera vez en la nueva Ley de Protección de Datos y Garantía de Derechos digitales [12]. El testamento digital se podría definir como el documento con todas las cuentas de correo, sitios webs, blogs, redes sociales, contraseñas, etc. Es decir, se trata de un documento donde se preverá que va a suceder con toda la información de carácter digital del testador. Este testamento digital ha de formalizarse ante notario, al igual que el testamento normal.

Cuando una ley similar sea aprobada en Argentina, los instrumentos musicales registrados en BlockGuitars podrán incluirse en el testamento digital de las personas.

3.3 Trabajos relacionados

Recientemente el Gobierno Nacional de Argentina ha certificado las ediciones electrónicas del Boletín Oficial mediante la utilización de la blockchain. De esta forma el Boletín Oficial adoptó un mecanismo adicional para que sus usuarios puedan verificar la autenticidad y obtener prueba de existencia de la edición electrónica.

En Argentina se desarrolló la herramienta Carnes Validadas, se trata de una herramienta tecnológica basada en blockchain que fortalece la trazabilidad de la carne roja desde la cría del animal hasta su consumo [14].

El municipio de Marcos Paz, en la provincia de Buenos Aires (Argentina), lanzó en 2019 la app ActivΦs mmp desarrollada por Koibanx. Se trata de una plataforma en la que conviven vecinos y comercios, donde se intercambian bienes y servicios usando activos digitales. Por su parte, Bolsas y Mercados Argentinos (BYMA) lanzó "BYMA Listadas", un nuevo sitio web con tecnología blockchain que permite a empresas emisoras realizar en forma remota múltiples trámites, como la solicitud de listado de nuevos valores negociables y la presentación de contenidos vinculados al régimen informativo [14].

También existen numerosas aplicaciones en otros países del mundo, por ejemplo España, Japón, Suecia y Reino Unido, están implementando la blockchain para registro de la propiedad sobre bienes raíces [15].

En resumidas cuentas, el sector público y privado puede tomar ventaja del blockchain y la velocidad en que los ecosistemas están creciendo alrededor de esta tecnología. Sin embargo, es natural que se encuentre un cierto temor que nace de la sensación de riesgo y retos que blockchain genera al introducir un cambio de paradigma en la forma de establecer los vínculos de confianza.

4 Conclusiones y trabajos futuros

Blockchain y los smart contracts son una novedosa tecnología que permite registrar de forma eficaz y segura transacciones de elementos virtuales, como criptomonedas. Pero también se puede aplicar a otros ámbitos. En este artículo se ha descripto la aplicación de esta tecnología para construir BlockGuitars [13], un sistema de registro de instrumentos musicales que ofrece mayor seguridad y ventajas que los sistemas actuales y que intenta ofrecer una solución a los problemas derivados de los frecuentes hurtos de estos elementos. La herramienta es inmutable, segura y distribuida.

El trabajo descripto en este artículo tiene dos aportes fundamentales, por un lado se describe la funcionalidad del sistema que está disponible y será de utilidad a los músicos argentinos, por otro lado se describe la metodología seguida para el desarrollo del mismo, así como también las decisiones de diseño adoptadas y la base tecnológica seleccionada. Estos elementos serán de utilidad a los desarrolladores que deban iniciar la construcción de un sistema similar en otro dominio.

En particular, se investigó la tecnología blockchain y todas sus ventajas, luego se continuó con un concepto muy importante como es smart contract y su lenguaje principal solidity. Para la construcción del prototipo funcional se utilizó truffle que es el framework más popular de solidity para realizar el smart contract, este se basó en el token ERC721 por su característica de ser no fungible, el framework vue.js para la parte del frontend y la librería web3.js para realizar la comunicación entre el smart contract y la parte visual.

Paralelamente se investigó acerca de la validez legar de estos registros virtuales, llegando a la conclusión de que los smart contracts son una evolución del sistema legal, no una sustitución del mismo y es necesario avanzar en la validez jurídica de los contratos y la estandarización en los distintos sectores. Uno de los principales retos de los contratos inteligentes es la dificultad de unir los tres mundos, el tecnoló-

gico, el legal y el dominio de aplicación. Los contratos inteligentes son escritos por los técnicos y los contratos propiamente dichos son escritos por los especialistas en leyes. Contar con expertos que dominen este nuevo paradigma es esencial para poder aprovechar al máximo todo lo que esta tecnología ofrece. Y en un futuro cercano, cuando las leyes se modernicen, tal como ha ocurrido en España y otros países, este tipo de registros virtuales podrán utilizarse no solo para registrar posesión sino también para formalizar temas de herencia y testamentos.

5 Referencias

1. Mercado clandestino de los instrumentos musicales. Diario La Voz. https://www.lavoz.com.ar/sucesos/el-mercado-clandestino-de-los-instrumentos-musicales Último acceso: Marzo 2020.
2. StolenGuitarRegistry http://www.stolenguitarregistry.com/ Último acceso: Marzo 2020.
3. PlayChecked https://www.playchecked.com/ Último acceso: Marzo 2020.
4. ScreamingStone http://www.screamingstone.com/ Último acceso: Marzo 2020.
5. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf . (2008).
6. Don Tapscott, Alex Tapscott Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, bussness, and the world.. Penguin Random House LLC. (2016).
7. Pastor J.: ¿Qué es el blockchain? https://www.xataka.com/especiales/que-es-blockchain-la-explicacion-definitiva-para-la-tecnologia-mas-de-moda . Septiembre 2018
8. Smart-contracts https://solidity.readthedocs.io/en/develop/introduction-to-smart-contracts.html . Último acceso Marzo 2020.
9. Ethereum. https://github.com/ethereum/wiki/wiki/Ethereum-Development-Tutorial. Último acceso Marzo 2020.
10. Solidity. https://github.com/ethereum/wiki/wiki/Ethereum-Natural-Specification-Format. Último acceso Marzo 2020.
11. Marchesi,M.,Marchesi,L.,Tonelli, R.: An Agile Software Engineering Method to Design Blockchain Applications, CEE-SECR '18: Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia Pages 1–8. (2018).
12. Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales, https://ayudaleyprotecciondatos.es/lopdgdd/ , ultimo acceso 11/06/2020.
13. Castelli Lluch, C.D., Tallarico, A.: BlockGuitars: una herramienta para las transacciones de instrumentos musicales mediante blockchain. Facultad de Informática, Universidad Nacional de La Plata. (2019)
14. Quirós, F.: Blockchain en Argentina: ¿Qué tan avanzado está su uso? https://es.cointelegraph.com/news/blockchain-in-argentina-how-advanced-is-its-use. Diciembre 2019.
15. Alcaide, J.: Blockchain para registro de la propiedad: países pioneros en su uso. https://blog.enzymeadvisinggroup.com/blockchain-registro-propiedad. Mayo 2019.