

Universidad Internacional de La Rioja (UNIR)

Escuela de Ingeniería

**Máster universitario en Dirección e Ingeniería de
Sitios Web**

RAMA PROFESIONAL

**Trazabilidad fiable
del vino:
*Blockchain***

Trabajo Fin de Máster

presentado por: Jiménez Godoy, Joaquín

Director: Boubeta Puig, Juan

El autor quisiera dedicar este trabajo a sus cincuenta años de actividad informática.
En el año 1967 trabajaba con computadores electromecánicos
basados en elementos discretos: transistores, resistencias, diodos...
En el año 2017 trabajaba con computadores en la nube.
Hoy, este Trabajo de Fin de Master (TFM) dedica unas páginas a sensores con elementos discretos,
y otras a la *blockchain* en la nube.
Ambos paradigmas están en los extremos de estos cincuenta años.
Pero este TFM también trata de otro mundo en permanente evolución: el vino.

Contenido

Contenido.....	5
Índice de figuras.....	8
Índice de tablas.....	9
Resumen.....	11
Abstract.....	11
1. Introducción.....	13
1.1 Motivación.....	14
1.2 Planteamiento del trabajo.....	15
2. Contexto y estado del arte.....	19
2.1 IoT productivo y predictivo.....	19
2.1.1 Bodega Pago Aylés.....	19
2.1.2 Ericsson.....	19
2.2 IoT para generar confianza.....	20
2.2.1 Ambrosus.....	20
2.2.2 My Story™.....	21
3. Objetivos y metodología de trabajo.....	23
3.1. Misión del TFM.....	23
3.2. Objetivos específicos.....	23
3.3. Metodología de trabajo.....	23
4. Aspectos de la trazabilidad del vino.....	25
5. Descripción de la arquitectura.....	27
5.1 IoT capa física del viñedo.....	28
5.1.1 Sensores.....	30
5.1.2 Plataformas de sensores del viñedo.....	31
5.1.3 Gateway Raspberry PI.....	40
5.1.4 Trabajadores.....	43
5.1.5 Clientes y distribuidores.....	43
5.2 Repositorio de la información.....	44

5.2.1 La Cadena de bloques	44
5.2.2 Propuesta de almacenamiento en la <i>blockchain</i>	45
5.2.3 Capa física de aporte de datos a la <i>side chain</i>	48
5.2.4 <i>Side chain</i> (pública con privilegios).....	49
5.2.5 <i>Main chain</i> (cadena principal y pública).....	51
5.2.6 <i>Scripts</i> de soporte temporales	53
5.3 WEB	54
5.3.1 Historia de temperatura en el viñedo	54
5.3.2 Lluvia y radiación solar	54
5.4.2 Tareas en el viñedo	54
5.4.3 Propuestas de representaciones gráficas responsivas	55
5.4 Historia de un vino MTB*	56
6. Desarrollo de un prototipo	59
6.1 Plataformas de sensores	59
6.1.1 Plataforma en viñedos.....	59
6.1.2 Plataforma en bodega	60
6.2 <i>Gateways</i>	60
6.2.1 <i>Gateway</i> en viñedo y estación meteorológica.....	60
6.2.2 <i>Gateway</i> en bodega	60
6.3 Dispositivos de los trabajadores	60
6.4 Dispositivos de los usuarios.....	60
6.5 Servidor	61
6.6 <i>Main chain</i> (pública).....	61
7. Líneas de trabajo futuras.....	63
7.1 Trazabilidad en bodega	63
7.2 Viaje de las botellas al consumidor	63
7.3 <i>Feedback</i> del consumidor	64
7.4 Correlación de eventos	64
8. Conclusiones.....	65

Anexo 1. Presupuesto plataforma sensores	67
Anexo 2. Presupuesto estación meteorológica.....	68
Anexo 3. Presupuesto <i>gateway</i>	69
Bibliografía	71

Índice de figuras

Figura 1. Ciclo del vino. Fuente OPENVINO	14
Figura 2. Propuesta de la trazabilidad del ciclo del vino	14
Figura 3. Lanzamiento de la criptomoneda MTB18 respaldada con vino.....	16
Figura 4. Propuesta de Ambrosus	21
Figura 5. Código QR para acceder a la trazabilidad My Story™ del vino	22
Figura 6. Arquitectura conceptual.....	27
Figura 7. Datos desde los sensores a la blockchain.....	29
Figura 8. Sensores, estación meteorológica y gateway	29
Figura 9. Vinduino en zona Petit Verdot fotografiado en otoño.....	31
Figura 10. Vinduino en detalle.....	32
Figura 11. Sensores de humedad en el viñedo	33
Figura 12. Lectura de los sensores de humedad. Fuente: Van der Lee Vineyard.....	35
Figura 13. Almacenamiento de la información. Fuente: Jordi Estapé	47
Figura 14. Gráfica de tareas y condiciones atmosféricas de un mes	55
Figura 15. Gráfica circular con datos tabulados para un año.....	56
Figura 16. Reflejo de un comercio justo	57
Figura 17. Maqueta para construir.....	59
Figura 18. El centro de este TFM	66

Índice de tablas

Tabla 1. Correspondencia entre el ID de la plataforma y el canal en ThingSpeak	42
Tabla 2. Tipos de blockchain. Fuente: McKinsey&Company	45
Tabla 3. Transacciones de tokens en la side chain	53

Resumen

El proceso de producción del vino no ha cambiado a lo largo de los siglos, y en él ha sido una constante el «tiempo» y sus circunstancias: vendimiar en el momento adecuado, retirar el mosto en el momento justo, regular la fermentación y envejecer el vino durante el tiempo necesario son «tempos» determinantes de la calidad del vino obtenido.

Por lo tanto, captar con sensores y actores todo lo ocurrido en el proceso del vino en cada «tempo» y almacenarlo para siempre en una *blockchain* (cadena de bloques) con minería de validación y certificación asegurará que se está ante la «verdad» de la vida de un vino.

En este Trabajo Fin de Máster (TFM) sobre la trazabilidad del vino, se definen métodos seguros y fiables de lectura del mundo físico basados en el Internet de las Cosas y se plantean procedimientos para almacenar información del proceso del vino en la *blockchain*, junto al gran reto de establecer una minería de la cadena de bloques generadora de confianza. En trabajos futuros se tratará el problema de cómo mostrar la información de la *blockchain* a grupos específicos de interés en la producción del vino, donde la información de la *blockchain* se complementará con la ayuda de oráculos.

Palabras clave: IoT, viñedo, sensores, vino, *blockchain*.

Abstract

The processes behind wine production have not changed that much over the centuries, where the importance of timing has always been the same: harvesting the grapes at the right time, beginning and ending fermentation at the right interval, and correctly ageing the final product.

As such, the timing of these processes, the lifecycle of wine production, can be measured by sensors, stored temporarily in the blockchain, and verified and certified by external actors and processes.

This Master Thesis defines secure and reliable methods for translating IoT (Internet of Things) data and human tasks from the viticulture and winemaking processes into a validated blockchain. Future efforts will build upon this data for non-repudiation and self-certification, presenting data to specific interest groups, where wine production data on the blockchain can be complemented through oracles.

Keywords: IoT, vineyard, grapes, sensors, wine, blockchain.

1. Introducción

En el libro *La Cuarta Revolución Industrial* (Schwab, 2016) se indica que el Internet de las Cosas (IoT) y la *blockchain* son dos tecnologías punteras —como también lo es el teléfono móvil— de una nueva época. Otros factores, como el continuo incremento del poder de computación, según establece la ley de Moore¹ y la caída de los precios del hardware, están haciendo posible conectar cualquier cosa a Internet, de tal manera que la información generada por estas cosas podrá ser procesada, distribuida y almacenada en diferentes tipos de bases de datos.

Por lo tanto, los objetos pasarán a ser inteligentes y podrán realizar dos funciones básicas: captación de datos (sensores) e interacción sobre activos (actuadores). Así que se puede sugerir, sin lugar a dudas, que en el futuro cualquier producto físico podrá conectarse a infraestructuras de comunicaciones ubicuas, y como consecuencia de ello sensores distribuidos por cualquier parte permitirán a los seres humanos o a otras máquinas percibir cualquier cosa que sea de su interés.

Por otro lado, la aparición de la *blockchain* (Yaga, 2018) como mecanismo de respaldo y de confianza para garantizar las transacciones realizadas con *bitcoin* ha permitido, al aplicar la *blockchain* a otros ámbitos diferentes de las criptomonedas, abrir unas expectativas aún por descubrir. Por ejemplo, la posibilidad de que el IoT envíe datos firmados digitalmente a la *blockchain*, donde serán almacenados de manera totalmente segura, permitirá disponer de una información veraz e inalterable del mundo que nos rodea. Esto abre nuevas posibilidades que solo hay que descubrir cómo aplicar (Don Tapscott, Alex Tapscott y Juan Salmerón, 2017).

Así que ambas cosas, el IoT en una de sus funciones aumentada (la captación de datos con auto certificación digital en origen) y la *blockchain* como base de datos que asegurará la integridad de la información y la guardará para siempre, podrán ofrecer transparencia sobre lo ocurrido a lo largo del ciclo de vida del vino: transparencia de lo ocurrido y realizado en el viñedo durante el cultivo de la uva, la fermentación de su prensado y la elaboración del vino en bodega, así como del almacenaje y distribución mayorista hasta la entrega final al consumidor. La Figura 1 resume el ciclo del vino que se deberá monitorizar.

¹ https://es.wikipedia.org/wiki/Ley_de_Moore

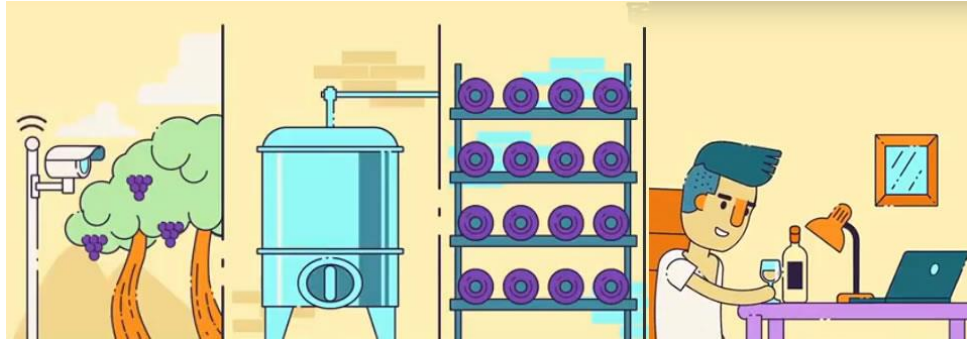


Figura 1. Ciclo del vino. Fuente OPENVINO²

Hoy se puede conseguir la transparencia y veracidad en la trazabilidad del vino durante las veinticuatro horas del día y todos los días del año, llegando a unos niveles de supervisión inalcanzables para las actuales entidades certificadoras. La figura 2 trata de mostrar cómo la existencia de sensores con identificaciones únicas, el envío de la información registrada y debidamente firmada, el almacenaje de esta información en la cadena de bloques, donde se asegura su integridad, y la presentación de cualquier detalle con técnicas web permite afirmar que estamos ante un nuevo paradigma de Internet.

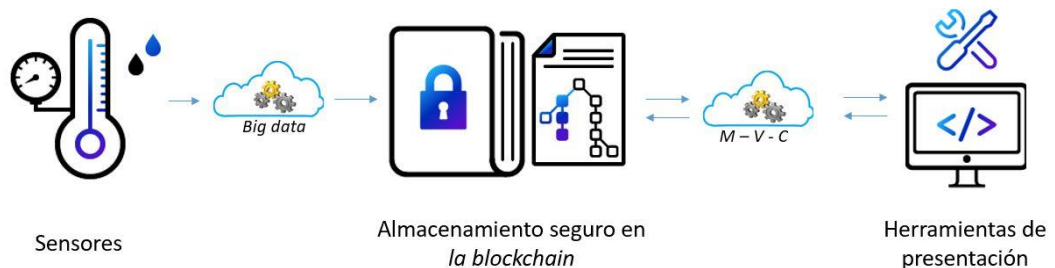


Figura 2. Propuesta de la trazabilidad del ciclo del vino

Este nuevo paradigma será consecuencia del internet de la *blockchain* o el internet de una sola verdad inalterable de la información, por lo tanto, del internet de la confianza entre las partes: proveedores y consumidores.

1.1 Motivación

La realización de este TFM nace de la voluntad por familiarizarnos con la *blockchain* y de aportar una propuesta novedosa al mundo del vino. Actualmente se acepta con resignación y

² https://youtube.com/watch?time_continue=2&v=7ZYISUzJBMo

como herencia de años que una etiqueta en el reverso de una botella de vino en la que se indica una denominación de origen es una garantía determinante de las características del producto que llega al consumidor. Nada más lejos de la realidad. Solo una captación fiable y permanente de todo lo ocurrido en el vino durante todo su ciclo de vida y una conservación de los datos recolectados en un repositorio permanente e inalterable puede garantizar la historia de un vino determinado.

Pese a la dificultad que pueda entrañar la implantación de este nuevo paradigma de la trazabilidad del vino, la tecnología que conforma la Cuarta Revolución Industrial invita a realizarlo. Es inevitable que esta novedad encuentre resistencias y oposiciones; pero así como el poderoso mundo que gestiona el comercio de los diamantes se ha resignado a delegar en la cadena de bloques la trazabilidad de cada uno de los diamantes producidos, evitando así la obtención de diamantes en origen con prácticas inhumanas (los llamados *blood diamonds*) o su tráfico ilegal, etc. (Marr, 2018), no cabe duda que en el mundo del vino, que también es una preciada joya, ocurrirá otro tanto y se optará igualmente por una trazabilidad transparente y aportadora de verdad, sobre todo cuando los ejemplos de aplicación de la *blockchain* sean algo cotidiano.

Por otro lado, este trabajo permite ampliar el espectro de aplicaciones del hardware libre basadas en Arduino para dotar de inteligencia a todo tipo de sensor y elevarlo a la categoría de objetos de Internet (IoT), así como aprovechar la potencia de servidores *low cost* como Raspberry Pi para realizar funciones de *gateway* de grupos de sensores, ya sea en el viñedo o en la bodega, enviando información a la nube para almacenar en una *blockchain* los datos generados por IoT que acompañan al vino durante su ciclo de vida.

Finalmente, la recopilación y correlación de información detallada, heterogénea y precisa sobre un viñedo permitirá determinar si el tipo de uvas, sus cuidados y la calidad del vino obtenido podría ser similar al producido en otro viñedo de gran reputación en cualquier otra parte del mundo, o simplemente determinar, lo que no es menos importante, si el vino obtenido de unas uvas determinadas de una añada será de la calidad que en condiciones óptimas puede alcanzar dicho vino o se ha quedado por debajo en un determinado porcentaje de la calidad esperada.

1.2 Planteamiento del trabajo

Este TFM se desarrollará alrededor del IoT del viñedo. Los datos recogidos por los sensores se almacenarán de tal manera que la cadena de bloques asegurará que nunca puedan alterarse ni los datos captados ni la información generada por los trabajadores. El resto de los apartados de la trazabilidad del ciclo del vino, como son el vino en la bodega o el viaje de las

botellas, recibirán un enfoque similar al propuesto para el viñedo y quedan para trabajos futuros.

En este caso nos proponemos realizar un trabajo de campo y otro de laboratorio. El trabajo de campo consistirá en desplazarse a un viñedo que disponga de una infraestructura basada en hardware y software libre. Se han localizado dos viñedos con estas infraestructuras: el viñedo del señor Van der Lee, en el valle de Temecula, California (Reinier van der Lee, 2016), donde hubo una iniciativa que originó una *start-up* que dio lugar a la plataforma sensórica Vinduino, y el viñedo del señor Barrow en Mendoza, Argentina, que también hace un uso extensivo de la tecnología Vinduino. Todo el código y documentación publicados sobre Vinduino están liberados de forma gratuita bajo formato *GNU General Public License 3.0*.

Sin embargo, el viñedo de Mendoza tiene un factor diferenciador para este TFM: el de ser la primera bodega en todo el mundo en vender su vino, MTB* (MIKETANGOBRAVO), en forma de criptomonedas (Barrow, 2018a), donde el propietario de un *token winecoin* MTB* es el propietario de una botella MTB. Esta característica hace que en este viñedo exista un gran conocimiento de la tecnología *blockchain*, que constituirá el núcleo de este TFM. La figura 3 muestra en qué fecha se produjo la *Initial Coin Offering* (ICO)³ de la cosecha del vino MTB del año 2018.



Figura 3. Lanzamiento de la criptomoneda MTB18 respaldada con vino

La existencia de *winecoins* MTB* (* es sustituido por dos números en el mundo real, 18 significa cosecha del año 2018, 19 significa año 2019, etc.) permitirá en la propuesta de

³ https://en.wikipedia.org/wiki/Initial_coin_offering

construcción de la *blockchain* para la trazabilidad del vino construir un paradigma de pago o compensación a los mineros, cada vez que estos logren realizar la validación de un bloque de información. Este aspecto se detallará en el apartado dedicado a la *blockchain*.

En una primera visita al viñedo Costaflores en Mendoza, se acordó con Mike Barrow implementar el TFM, haciendo uso de los datos que permanentemente genera la infraestructura IoT del viñedo. También se valoró y se planificó construir una réplica de la infraestructura, que diera una respuesta capaz de monitorizar el ciclo de vida del vino, para en un futuro avanzar en el objetivo de lograr una trazabilidad del vino completa y fiable.

Actualmente, la información captada del viñedo Costaflores es enviada a Thingspeak⁴ (plataforma abierta para el IoT en combinación con la analítica de MATLAB), para la construcción de gráficas en tiempo real con los datos recogidos en el viñedo. Por lo tanto, habrá que actualizar los programas del *gateway* para que la información, además de ser enviada a Thingspeak, sea enviada también a un repositorio donde la información sea almacenada y asegurada con procedimientos de *blockchain*, para así asegurar la integridad de la información, su no repudio y, como se verá más adelante, su alta disponibilidad.

Durante la estancia en Costaflores también se instaló un servidor Linux con sistema operativo CentOS 7 en un hardware NUCIntel, lo que permitirá gestionar la red interna del viñedo remotamente desde Barcelona o desde cualquier otro punto del planeta, con la finalidad de realizar las actualizaciones que vaya requiriendo el software en el viñedo. También se configuró un arranque automático del sistema NUCIntel, para que, en caso de fallo de la acometida eléctrica del viñedo y su posterior recuperación, el sistema pueda realizar un arranque automático.

Para poder tomar el control remoto de la infraestructura del viñedo, se instaló el cliente Teamviewer⁶ sobre un servidor NUCIntel, si bien esta instalación resultó de una dificultad no esperada. El sistema operativo gestionando el servidor NUCIntel es CentOS 7 y la instalación de Teamviewer en CentOS 7 requiere un número importante de dependencias. Finalmente, la instalación de Teamviewer se realizó con éxito y el procedimiento para llevarla a cabo está descrita en la wiki de Openvino⁷. El apartado de la wiki que describe esta instalación tiene el nombre Foxtrot, que es como se bautizó el conjunto formado por el hardware NUCIntel más el sistema operativo CentOS 7 .

⁴ <https://thingspeak.com/channels/311015>

⁶ <https://www.teamviewer.com/es/>

⁷ <http://wiki.costaflores.com:8090/display/OP/foxtrot>

Hay que destacar que las plataformas de sensores del viñedo están alimentadas con energía solar y dotadas de baterías de respaldo, las cuales se cargan también utilizando la energía solar. También existe en el *gateway* la posibilidad de almacenar de forma permanente o temporal la información generada en el viñedo en una BBDD SQLite. De esta manera se asegura que los datos en su tránsito por el *gateway* no se pierdan en periodos de fallo de acceso a Internet y más tarde se puedan reenviar los datos cuando se restablezca la conexión a Internet.

El trabajo en laboratorio, cuando se realice, además de implementar todos los componentes básicos y necesarios para poder reproducir en una maqueta el IoT —el hardware y software del viñedo—, implantará la conexión de las plataformas de sensores vía una red LoRaLAN para la transferencia de datos a la puerta de salida —*gateway*— Raspberry PI y el envío de estos datos desde el *gateway* vía Internet a la *blockchain* para su almacenamiento. Son estos datos almacenados los que permitirán, posteriormente, construir la trazabilidad del vino con el alcance funcional que se desee.

2. Contexto y estado del arte

En este capítulo se presenta el estado del arte en implementaciones similares. Como se verá ya existen implantaciones de captación de datos en el viñedo, pero no soluciones de almacenamiento de los datos captados en sistemas de almacenamiento que aseguren su durabilidad, su disponibilidad en todo momento y la confianza en la información almacenada.

Pero, sobre todo, no existen proyectos vinícolas que ofrezcan transparencia del producto que elaboran, basados en la verdad, y que pongan esta información a disposición de los consumidores con procedimientos de acceso universales y sin ningún tipo de censura.

2.1 IoT productivo y predictivo

Durante los años previos a este trabajo, en gran medida desde el año 2010, han ido apareciendo soluciones de IoT en el viñedo —por ejemplo, soluciones Ericsson—, pero todas ellas enfocadas a recopilar datos para ganar en eficiencia productiva, ya sea en la cantidad de uva cosechada, en su calidad o en la reducción de costes. También se viene utilizando el IoT del viñedo para fines predictivos —soluciones Libelium—, para así poder anunciar con suficiente antelación la posible llegada de plagas, gracias a patrones confeccionados sobre análisis forenses de datos recopilados previamente, o predecir comportamientos climatológicos muy locales y ajustados a la superficie del viñedo.

Entre los casos de éxito publicados sobre el paradigma —productividad y predicción— descrito anteriormente, pero contruidos sobre soluciones de hardware y software propietario, merecen ser destacados los siguientes casos.

2.1.1 Bodega Pago Aylés

En la bodega Pago de Aylés (CIO, 2018), las empresas remOT Technologies y Libelium han desplegado una red mallada de 100 puntos de captación de datos con 25 parámetros de medida cada uno. La obtención de datos de forma continua está permitiendo establecer patrones de comportamiento y establecer modelos predictivos.

2.1.2 Ericsson

La solución de Ericsson *Device Connection Platform* (Plataforma de Conexión de Dispositivos) (Hernández, 2016) ha permitido a los enólogos del Valle del Mosel en Alemania instalar la solución Tracovino, que monitoriza datos esenciales para mejorar la calidad del vino. Esta solución dentro de Ericsson se conoce como el «Internet de las Uvas» y hará innecesario tener que mirar al cielo para saber si lloverá o hará sol, o echar mano de experiencias anteriores para conocer el mejor momento para recoger la cosecha.

2.2 IoT para generar confianza

El IoT colaborador en la trazabilidad del vino será visto como el servicio inferior de una estructura de servicios en capas cuya finalidad es la de transmitir confianza a los consumidores. La capa básica o inferior de este conjunto de servicios generadores de confianza podrá incluir también las mismas prestaciones que el IoT productivo y predictivo. En estos momentos esta propuesta no es algo aislado, ya que durante la elaboración de este trabajo se han detectado algunas iniciativas que comparten algunos de sus objetivos, si bien todas ellas parecen estar en fase de iniciación. A continuación se seleccionan dos de las soluciones más impactantes encontradas.

2.2.1 Ambrosus

El ecosistema en desarrollo por Ambrosus (Ambrosus, 2018), si bien se pretende aplicar en general para cualquier alimento o medicina, permite hacer una abstracción de su marco de implantación para compararla con la trazabilidad de confianza del vino aquí propuesta.

La siguiente imagen muestra como Ambrosus propone a grandes rasgos protocolizar el IoT generador de información para la trazabilidad de cualquier alimento o medicina a lo largo de la cadena de suministros. Ambrosus genera un identificador único para cada entidad (producto, lote, caja, pallet, contenedor, etc.) de la que se desea obtener la trazabilidad, es decir, su seguimiento y análisis de propiedades a lo largo de su ciclo de vida. Esta entidad es almacenada con toda su historia de forma inmutable en la cadena de bloques de Ambrosus.

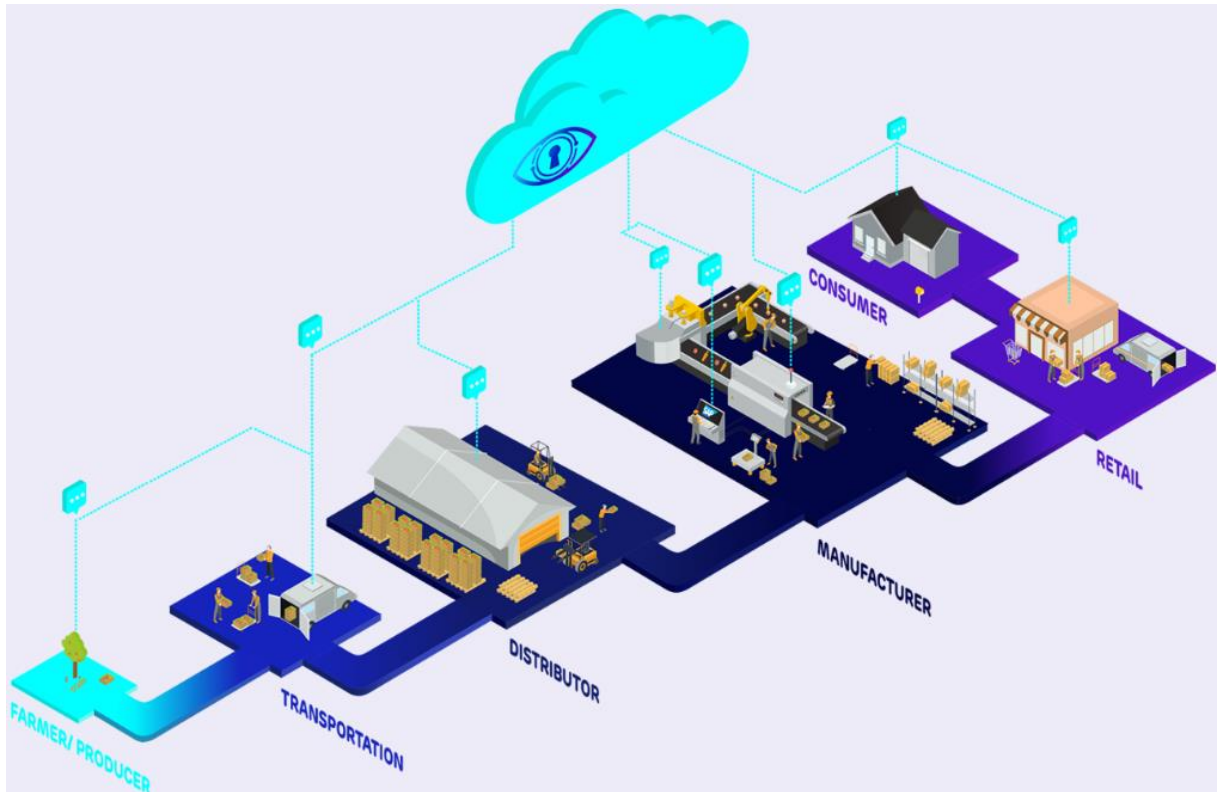


Figura 4. Propuesta de Ambrosus

Y lo concretan diciendo: «*Blockchain* almacena los datos una vez y los conserva para siempre. Pero ¿qué hay de la calidad de los datos? Ambrosus combina *blockchain* con un sistema de sensores para crear una solución de integridad de los datos de extremo-a-extremo en cualquier proyecto dentro de cualquier corporación. Ambrosus convierte los sensores en oráculos⁸ de reputación. Es decir, una fuente de datos verdaderos para incorporar a libros de contabilidad distribuidos.»

2.2.2 My Story™

My Story™ es una de las aplicaciones desarrolladas por la empresa DNV GL (Rrisciotti, 2018) en combinación con VeChain. Si bien es difícil encontrar información detallada acerca de la infraestructura tecnológica que se encuentra detrás de esta solución, sí que se puede extraer su parecido con el objetivo de este TFM, ya que en unas declaraciones a la prensa los directivos de DNV afirmaban que My Story es el primer paso en un camino mucho más largo en el que se aprovechará el uso combinado y coherente de la *blockchain*, el IoT, la analítica de los datos y la inteligencia artificial, con el fin de ayudar a los productores a crear confianza en sus productos, mejorar sus prestaciones y aumentar la eficiencia en los procesos.

⁸ blog.ledger.co/2016/08/31/hardware-pythias-bridging-the-real-world-to-the-blockchain/#.2zegzh6f

Las personas que tengan la posibilidad de disfrutar de un vino bajo el paraguas de esta tecnología, solo tendrán que acceder a la plataforma de datos Veracity de DNV GL, vía el código QR situado encima de la etiqueta trasera del vino (véase la figura 5).



Figura 5. Código QR para acceder a la trazabilidad My Story™ del vino

3. Objetivos y metodología de trabajo

El objetivo principal de este TFM es realizar una propuesta tecnológica e innovadora para conseguir la integridad y disponibilidad de datos del IoT generados en el viñedo, la bodega y el transporte de las botellas hasta el consumidor final. Dicho objetivo se alcanzará con la convergencia de diferentes tecnologías como son el IoT, la firma digital, los servicios web, la *blockchain*, los oráculos, las apps y la web.

El alcance de este TFM se limita a los datos generados en el viñedo Costaflores. Cuando se termine este TFM los trabajos continuarán y se seguirán publicando en un apartado de la wiki OPENVINO (Barrow, 2018b), que ahora es el repositorio de trabajos recopilados para esta actividad.

3.1. Misión del TFM

Este trabajo tiene la misión de definir una arquitectura de una plataforma abierta y viva para publicar los datos del viñedo e iniciar las tareas de su construcción. Si algún visitante tiene una inquietud determinada sobre los datos almacenados y la respuesta no está construida sobre la salida estándar inicial (web), la plataforma permitirá su construcción al aportar APIs para facilitar la interacción de apps que den respuestas a enólogos, críticos, viticultores, etc.

3.2. Objetivos específicos

Los objetivos específicos de este TFM son definir una arquitectura software que permita:

- Emitir datos desde el mundo físico a un repositorio centralizado
- Almacenar y validar los datos recibidos en la cadena de bloques
- Mostrar un esquema básico de datos vía web

3.3. Metodología de trabajo

Realización de entrevistas a expertos en las materias principales de este proyecto, como son:

- Conocedores de las funcionalidades alrededor de la elaboración del vino
- Técnicos con el conocimiento de ubicación de la sensórica en el recorrido de la producción del vino
- Conocedores de las diferentes implantaciones de la *blockchain*
- Especialistas en mostrar datos en la web relacionados con la industria del vino

Estas entrevistas han dado lugar a publicaciones con extractos de ellas en la wiki de Openvino⁹, así como a la publicación, en la misma wiki, de algunos trabajos individuales o realizados conjuntamente con las personas entrevistadas.

Dentro de la metodología de trabajo ha estado el convivir unos días en el viñedo Costaflores, de Mendoza, para adquirir formación en el cuidado de un viñedo y conocer las posibles implantaciones de soluciones de sensórica en ese ámbito. Aprovechando la visita se instaló un servicio que permite el acceso remoto al viñedo, para poder realizar así actualizaciones de software que permitan el envío de la información de los datos a la *blockchain*.

Asimismo, se establecen diferentes sesiones de trabajo para implantar la propuesta de este TFM a medida que se avanza en su definición, que será una de las actividades futuras de trabajo, como se verá en esta memoria.

Como resultado de estos trabajos se ha definido una propuesta de arquitectura que combina la captura de datos, su almacenamiento permanente e inalterado en la *blockchain*, así como la visualización de la trazabilidad del vino.

⁹ [http://wiki.costaflores.com:8090/display/OP/Publishing+Vineyard+Data Internet](http://wiki.costaflores.com:8090/display/OP/Publishing+Vineyard+Data+Internet)

4. Aspectos de la trazabilidad del vino

Según detalla Josep Forroll (Forroll, 2018), la trazabilidad del vino, basada en información registrada manual o automáticamente y almacenada en una cadena de bloques, de forma inalterable y segura, integra los siguientes aspectos:

- La uva y su cuidado
 - Tipo de terreno (*terroir*), marco de plantación.
 - Variedad de la uva, edad de la cepa
 - Condiciones climáticas, en especial sol, pluviometría y momentos en que se producen
 - Humedad del suelo a diferentes profundidades (entre 0,05 metros y 2 metros)
- Tareas en el viñedo
 - Grado de azúcar de la uva
- Recolección
 - Sistema de vendimia: manual o mecánico
 - Momento de la vendimia: de día o de noche
 - Temperatura de recolección
 - Método y duración del traslado de la uva vendimiada hasta las prensas
- Trabajo en bodega
 - Elaboración
 - Tipo de prensa
 - Porcentaje de extracción
 - Temperatura de fermentación
 - Tipo de depósito durante la fermentación
 - Tratamientos físicos durante el proceso de elaboración
 - Posibles segundas fermentaciones (malolácticas)
 - Sistemas de estabilización y clarificación
 - Posibles *coupages*
 - Envejecimiento
 - Tiempos y tipo de guarda
 - Tipo y tiempo en bodega (crianza)
 - Sistemas de llenado
 - Cambio y vaciado de barricas
- Embotellado
 - Sistema de traslado del vino
 - Gravedad

- Bombas
- Tipo de tapón
- Envejecimiento en botella
 - Condiciones de temperatura y humedad de la bodega
- Transporte de las botellas
 - Cambios de presión
 - Cambios de temperatura.
 - Traqueteos
 - Exposiciones al sol
- Almacenado de las botellas antes de entregar al consumidor
- Opinión del consumidor

5. Descripción de la arquitectura

La arquitectura que se propone radica en integrar diferentes tecnologías ya en funcionamiento y consolidadas, para contribuir a abrir un proceso de valor en Internet, aportando una trazabilidad del vino bajo estándares de tecnologías de no repudio y de integridad de la información.

Esta trazabilidad del vino podrá exportarse a otros campos, generalizándose el concepto de trazabilidad inalterable y de confianza en su aplicación a todo el ciclo de vida de alimentos, que va desde la obtención de materias primas, su posterior elaboración para ser transformadas en bienes de consumo y su final distribución al consumidor.

La Figura 6 muestra una visión general y conceptual de la arquitectura que se propone en este ámbito.

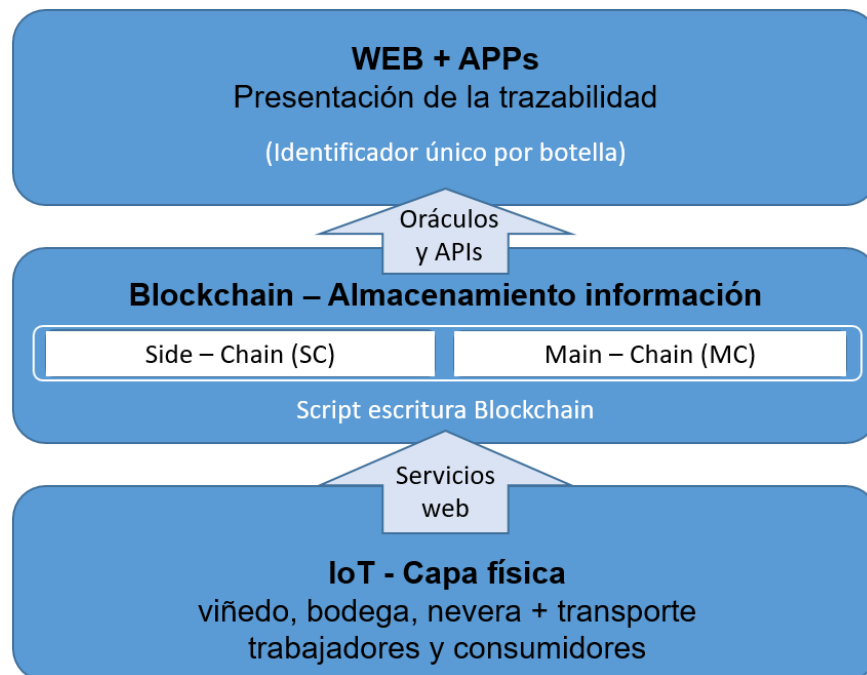


Figura 6. Arquitectura conceptual

Y las funciones básicas de cada capa son:

1. **IoT – Capa física.** Aplicar extensivamente protocolos IoT para obtener datos de los sensores, como pueden ser *OneWire*¹⁰ e IIC¹¹ (Inter-Integrated Circuit, I²C), para su envío cíclico al *gateway* vía LoRaLAN —módems sin hilos de largo alcance vía radio¹²— y posterior envío de los datos al repositorio de datos central vía servicios web

¹⁰ <https://en.wikipedia.org/wiki/1-Wire>

¹¹ <https://en.wikipedia.org/wiki/I%C2%B2C>

¹² http://www.globalsat.com.tw/files/LM-130%20specification_Ver-1.5_20161207.pdf

a través de *gateways*. También son fuentes de datos en esta capa física los trabajadores y consumidores.

2. **Blockchain – Almacenamiento de la información.** Asegurar con *blockchain* la integridad de los datos obtenidos desde la capa inferior y permitir que se pueda acceder a estos datos desde la capa superior.
 - La *blockchain* en esta propuesta será una combinación de dos cadenas de bloques: la *Side Chain* (SC), que será pública con control de accesos —*permissioned*¹³— y la *Main Chain* (MC), que será pública sin control de accesos —*permissionless*¹⁴—. Esta propuesta se verá en más detalle en el apartado *blockchain* de este TFM.
3. **Web + oráculos + apps.** Herramientas de presentación de información y extracción de los datos del repositorio central vía APIs. Se utilizará como enlace a la información el código QR (*Quick Response Code*) de cada botella. En el momento de acceder a la *blockchain* con ayuda de oráculos, se podrá completar la información básica con información extra, como puede ser la cotización de un *winecoin* MTB* en el mercado de intercambios (*exchange*).

5.1 IoT capa física del viñedo

En este apartado se definirá el funcionamiento de la monitorización del viñedo. Esto se conseguirá con la lectura de los datos generados por la sensórica, dentro del paradigma del IoT, con la aportación adicional de información sobre las tareas realizadas por los trabajadores y con las capturas de datos no automatizadas, es decir, realizadas manualmente con métodos específicos. Estas capturas de datos en su totalidad están dedicadas a monitorizar las actividades relacionadas con la crianza de la uva, la calidad alcanzada por ella y su recolección, así como a asegurar que los envíos de todos los datos generados en el viñedo lleguen a la cadena de bloques.

La Figura 7 muestra una visión general de la toma de datos en el viñedo y su envío a la *blockchain*.

¹³ <https://www.investopedia.com/terms/p/permissioned-blockchains.asp>

¹⁴ <https://graylinegroup.com/responsible-experimentation-with-blockchain/>

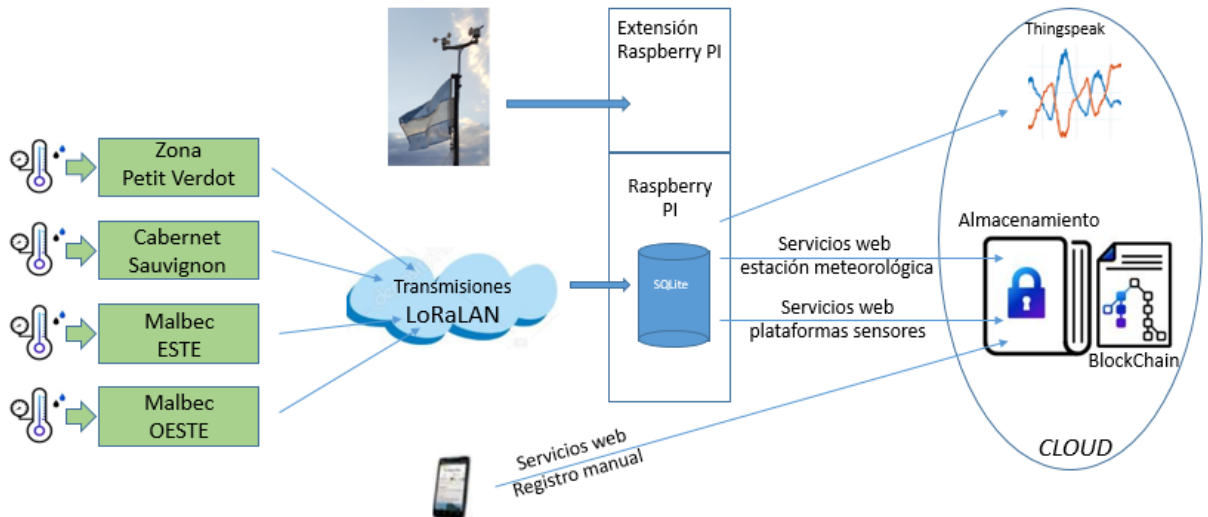


Figura 7. Datos desde los sensores a la blockchain

En el diagrama anterior, correspondiente a nuestro caso de estudio, se muestran a la izquierda las plataformas concentradoras de sensores para mantener monitorizadas zonas parciales del viñedo no mayores de una hectárea. Para su mejor identificación, estas zonas han sido bautizadas con el nombre del tipo de uva cultivada en cada área: Petit Verdot, Cabernet Sauvignon, Malbec Este y Malbec Oeste. En la figura 8 se muestra la ubicación en el viñedo de las plataformas de sensores y del gateway con el que se comunican unidireccionalmente vía LoRaLAN

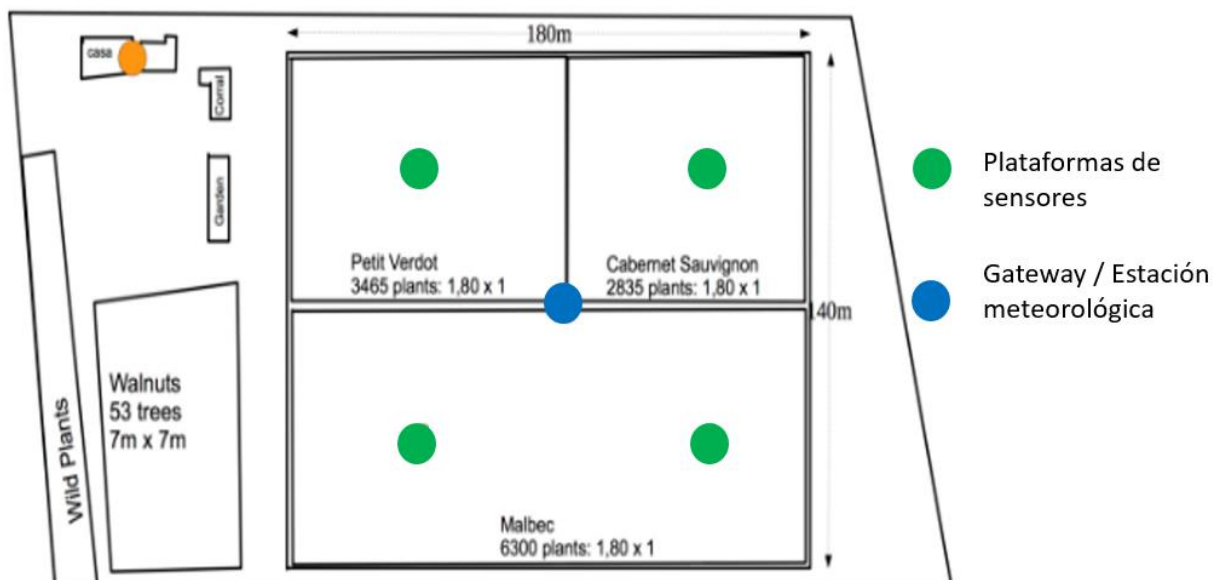


Figura 8. Sensores, estación meteorológica y gateway

En el centro del viñedo se encuentra la estación meteorológica, conectada a la placa de extensión del sistema Raspberry Pi. Con ella se recogen datos generales del viñedo. El sistema Raspberry Pi es la puerta de salida o *gateway*, que tiene la responsabilidad de enviar todos los datos captados por la sensórica del viñedo a la *blockchain*, sistema central de almacenamiento de la información.

Por último, están los datos que se toman manualmente, como el grado de azúcar de la uva. Estos datos manuales se envían al sistema de almacenamiento vía comunicaciones móviles.

5.1.1 Sensores

En este trabajo se propone que las plataformas de sensores estén basadas en sistemas Arduinos evolucionados (Vinduinios) en lo que respecta a la PCB (*Printed Circuit Board*) para así eliminar las placas de extensión *proto-boards*. Solo se consideran en este TFM aquellos sensores que aportan datos de interés para la trazabilidad del vino definida en las funcionalidades sobre la producción del vino que en este trabajo se han expuesto, prescindiendo de aquellos sensores que pueden recoger datos específicos con finalidades predictivas, por ejemplo, para la detección de plagas. Sensores que aportan información que debe ser correlacionada con técnicas de *big data* para descubrir patrones disparadores de acciones. Aunque la característica predictiva queda fuera de este trabajo, sí que será de interés en trazabilidades futuras registrar las plagas o amenazas de plagas que haya sufrido el viñedo durante la producción de una añada en particular.

La sencillez de acoplar a los Arduinos todo tipo de sensores, ya sea directamente a la propia placa Arduino o a las placas de expansión conocidas como *proto-board*, así como la existencia de protocolos que permiten la lectura de todo tipo de sensores desde Arduino (por ejemplo, los protocolos *OneWire*) permiten dar respuesta a la necesidad de controlar aspectos como:

- Temperatura ambiente
- Humedad ambiente
- Presión atmosférica
- Pluviometría
- Velocidad del viento
- Dirección del viento
- Radiación ultravioleta
- Radiación solar
- Temperatura del suelo
- Humedad del suelo, 4 profundidades.
- Humedad de las hojas

- Diámetro del tallo y de los frutos (uvas)

Siguiendo las directrices de los expertos, habrá dos clases de sensores en función de su lugar: los que se colocarán en diversos lugares del viñedo y los que tendrán una única ubicación dentro del viñedo. Por ejemplo, sensores para la detección de la humedad del terreno se realizará en múltiples ubicaciones, y sensores para medir la pluviometría se instalarán en un solo lugar.

5.1.2 Plataformas de sensores del viñedo

Las plataformas de sensores, Vinduinos, se distribuyen por todo el viñedo (como mínimo una por hectárea), estructurando una red (IoT) que da cobertura a la captación de datos denominados locales.

Estas plataformas locales están basadas en hardware y software Arduino. El hardware ha sido modificado para no necesitar las placas de expansión *proto board* ni las típicas placas *shield*. En la figura siguiente se muestra un Vinduino instalado en una casita para pájaros, que permite instalar fácilmente la placa solar y mantener la electrónica aislada de las inclemencias externas.



Figura 9. Vinduino en zona Petit Verdot fotografiado en otoño

Para una mayor familiarización con la plataforma de sensores, su funcionamiento y los datos recogidos, a continuación se detalla brevemente la solución instalada actualmente en la finca Costaflores. Los datos recogidos por las plataformas de sensores, alimentadas con energía solar y baterías de respaldo, llegarán al *gateway* vía módems LoRaLAN. A partir de ese

momento se procederá a realizar determinados sistemas de redundancia de los datos, para asegurar que estos siempre lleguen a la *blockchain* pese a los fallos de acceso a Internet.

La figura 10 muestra una descripción de la plataforma de sensores Vinduino R3 y a continuación se detallan las diferentes partes.

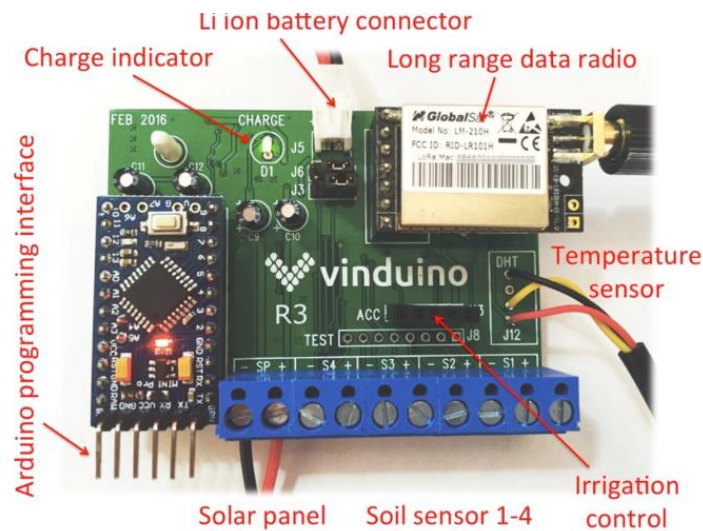


Figura 10. Vinduino en detalle

Panel solar (monitorizado por el LED indicador de carga). La plataforma remota está concebida para el envío de los datos asociados a sus sensores cada 15 minutos (más un tiempo adicionado al azar para evitar colisiones con los envíos de datos de otras plataformas de sensores), durante las 24 horas del día. Para suministrar la energía requerida para que el Vinduino realice las funciones detalladas anteriormente, este dispone de un panel solar que durante las horas de sol mantiene cargada una batería. El voltaje de la batería también es medido y enviado en todos los ciclos en los que se envía la información de los sensores. Un voltaje de la batería inferior a 3,6 V informa de posibles problemas en la alimentación del Vinduino y aconseja que la batería sea reemplazada.

La línea de código utilizada en el Vinduino para medir el voltaje es:

```
float Vsys = analogRead(3)*0.00647; // read the battery voltage
```

Dentro de cada bucle de ejecución del software en el Vinduino, el voltaje de la batería se almacenará en la variable *Vsys*, para ser enviado posteriormente, en nuestro caso al *gateway*, para su posterior almacenamiento en la *blockchain*.

Timer. Los Vinduiños, para un mayor aprovechamiento energético, están dormidos durante 15 minutos. Pasado este tiempo son activados, ejecutan la lectura de todos sus sensores asociados y en el mismo ciclo de lectura envían los datos capturados al *gateway*. Una vez realizada la tarea de lectura de los sensores y envío de la información, vuelven al estado de no actividad durante otros 15 minutos.

Los 15 minutos se establecen en la función `settimerPCF8563()`:

```
#include <Wire.h> // I2C communication to RTC
#define PCF8563address 0x5 // Format data for RTC

void settimerPCF8563() // this sets the timer from the PCF8563
{
  Wire.beginTransmission(PCF8563address);
  Wire.write(0x0E); // move pointer to timer control address
  Wire.write(0x83); // sends 0x83 Hex 010000011 (binary)
  // (0x82 = enable timer, timer clock set to 1 second)
  // (0x83 = enable timer, timer clock set to 1 minute)
  Wire.endTransmission();

  Wire.beginTransmission(PCF8563address);
  Wire.write(0x0F); // move pointer to timer value address
  Wire.write(0x0F); // sends delay time value (0F =15 minutes, 3C = 1 hour)
  Wire.endTransmission();
}
```

Sensores. Cada plataforma Vinduiño tiene conectados 4 sensores de humedad que penetran en el terreno a profundidades de 0,05, 0,5, 1 y 2 metros. Especial relevancia en este TFM tiene el sensor número 1, situado a 2 metros de profundidad. Se sabe que la humedad a dos metros de profundidad no es de utilidad para las raíces de la cepa. Por lo tanto, para mostrar el compromiso o la solidaridad de un viñedo con la escasez del agua (WWF, 2018), este sensor deberá marcar no humedad en periodos en que la estación meteorológica no denuncie lluvias. Con la ayuda de Figura 11 se describe el porqué de la existencia de 4 sensores.

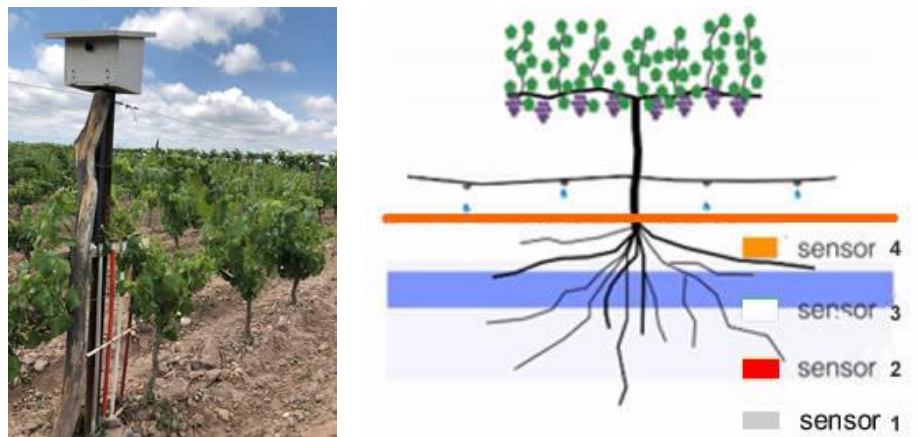


Figura 11. Sensores de humedad en el viñedo

El sensor 2 tiene la responsabilidad de disparar el corte del riego relativo a la zona por él monitorizada y lo hace ordenando el cierre de la válvula de riego. Por otro lado, como el estado de los cuatro sensores es enviado a la *blockchain* cada 15 minutos, más un tiempo definido al azar, de este modo no solo se garantiza el uso responsable del agua, sino que también se certifica el grado de humedad del que disfrutaban las raíces de la cepa durante todo el ciclo productivo o de actividad de la cepa.

En las plataformas de sensores, la gestión —lectura— de los sensores de humedad no sigue ningún protocolo de los reconocidos como estándares; por lo tanto, a continuación se describe su funcionamiento básico.

El código Arduino puede forzar fácilmente cambios de ON-OFF en dos de sus salidas digitales, logrando simular el suministro de corriente alterna a los sensores de humedad, como se verá más adelante. De este modo, a la par que se excitan los sensores de humedad para poder medir la misma, bajo efectos parecidos a la circulación de corriente eléctrica, se evita el deterioro de los electrodos, como ocurriría si solo se aplicara corriente continua. Los diferentes valores de la intensidad de la corriente eléctrica que circula por los circuitos que se establecen determina diferentes valores de caídas de tensión en las resistencias conocidas (*knownResistor*) con las que se forma un circuito en serie para cada sensor. El circuito se ve afectado por la conductividad eléctrica en mayor o menor medida dependiendo de la humedad del terreno que rodea cada sensor. Estas caídas de tensión en los circuitos establecidos con cada sensor son leídas por la plataforma de sensores y convertidas en datos que se almacenan en unas variables determinadas.

A continuación es posible simular lo expuesto anteriormente y entender el fenómeno para un solo sensor. En la siguiente figura se muestra un ejemplo basado en Arduino que describe el funcionamiento de un sensor de humedad del suelo.

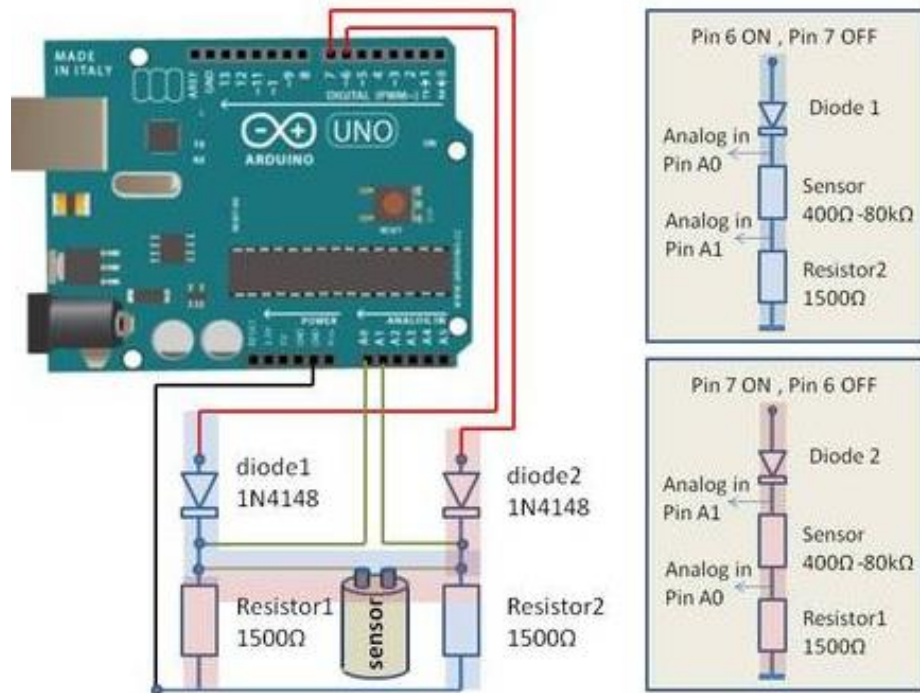


Figura 12. Lectura de los sensores de humedad. Fuente: Van der Lee Vineyard

Para interpretar el ejemplo de la figura anterior hay que remontarse a la asignatura de física en sus capítulos relacionados con la electricidad y recordar las leyes de Kirchoff. Una de sus leyes establece que, ante la circulación de corriente eléctrica a través de dos resistencias en serie (en este caso están en serie el sensor, que equivale a una resistencia variable, y una resistencia conocida con un valor fijo), se produce una caída de tensión entre los extremos de las dos resistencias, y esta caída de tensión se divide de forma proporcional a los valores de la resistencia óhmica de cada componente.

El seguimiento del circuito establece que dependiendo de los valores ON/OFF en los pines digitales lógicos 6 y 7 del Arduino, la corriente eléctrica circulará unas veces por el diodo D1 y otras por el diodo D2, y los cambios de sentido de la corriente continua producirán un efecto de corriente alterna sobre el sensor. Como consecuencia de ello, las entradas analógicas A0 y A1 del Arduino irán leyendo los valores de la tensión «caída» en el sensor. Esta tensión será de unos valores u otros dependiendo del grado de humedad del sensor, y la tensión será de signo positivo o negativo en función del sentido de la corriente eléctrica en el sensor.

Ya en el caso real que nos ocupa, en el Vinduino, con la ayuda de un multiplexor se irán excitando todos los sensores uno tras otro, y el cálculo resultante de medir varias veces cada sensor y promediarlo se irá acumulando en las variables `read1..read4`. El contenido de las variables `readn` se enviará posteriormente al *gateway*. El código para leer los valores de cada sensor es el siguiente:

```

#include <math.h>

/// Creando el entorno para leer los 4 sensores varias veces en cada ciclo.
#define NUM_READS 10 // Número de lecturas de cada sensor
const long knownResistor = 4750; // Valor en ohmios
unsigned long supplyVoltage; // Valor del voltaje suministrado
unsigned long sensorVoltage; // Valor del voltaje medido en el sensor
int zeroCalibration = 115; // Reajuste del circuito en caso de
cortocircuito.
long buffer[NUM_READS];
int index2;
int i; // Variable para indexar
int j=0; // Variable para indexar

void setup() {

// configuración de la interface de los sensores

// Inicialización de los pines D5, D6 como entradas de alta impedancia.
// Pines D5, D6 son para excitar los sensores
pinMode (5, INPUT);
pinMode (6, INPUT);
// Pin 7 es para habilitar los circuitos del MUX
pinMode (7, OUTPUT);
// Pin 8,9 son para seleccionar los sensores del 1 al 4
pinMode (8, OUTPUT); // Entrada A del MUX
pinMode (9, OUTPUT); // Entrada B del MUX

soilsensors();
...

void loop()
...
void soilsensors() {
// Selección del circuito del sensor 1, habilitación en el MUX y lectura del
sensor.

digitalWrite(8, LOW);
digitalWrite(9, LOW);
digitalWrite(7, LOW);
measureSensor();
unsigned long read1 = average();

// Selección del circuito del sensor 2, habilitación en el MUX y lectura del
sensor.

digitalWrite(8, LOW);
digitalWrite(9, HIGH);
digitalWrite(7, LOW);
measureSensor();
unsigned long read2 = average();

// Selección del circuito del sensor 3, habilitación en el MUX y lectura del
sensor.

digitalWrite(8, HIGH);
digitalWrite(9, LOW);
digitalWrite(7, LOW);
measureSensor();
unsigned long read3 = average();

// Selección del circuito del sensor 4, habilitación en el MUX y lectura del
sensor.

digitalWrite(8, HIGH);
digitalWrite(9, HIGH);
digitalWrite(7, LOW);
measureSensor();

```

```

    unsigned long read4 = average();
}

...

void measureSensor() {

    for (i=0; i<NUM_READS; i++) {

        pinMode(5, OUTPUT);
        digitalWrite(5, LOW);
        digitalWrite(5, HIGH);
        delayMicroseconds(250);
        sensorVoltage = analogRead(0); // leer la tensión en el sensor
        supplyVoltage = analogRead(1); // leer el voltaje suministrado
        digitalWrite(5, LOW);
        pinMode(5, INPUT);
        long resistance = (knownResistor * (supplyVoltage - sensorVoltage ) /
        sensorVoltage)-zeroCalibration ;

        addReading(resistance); // Adiciona las lecturas
        delayMicroseconds(250);

        pinMode(6, OUTPUT);
        digitalWrite(6, LOW);
        digitalWrite(6, HIGH);
        delayMicroseconds(250);
        sensorVoltage = analogRead(0); // Lee el voltaje en el sensor
        supplyVoltage = analogRead(1); // Lee el voltaje suministrado
        digitalWrite(6, LOW);
        pinMode(6, INPUT);
        long resistance = (knownResistor * (supplyVoltage - sensorVoltage ) /
        sensorVoltage)-zeroCalibration ;

        addReading(resistance); // Adiciona las lecturas
        delayMicroseconds(100);

    }

}

...
// Algoritmo para almacenar en el array buffer las lecturas sobre cada uno de los
sensores

void addReading(long resistance) {
    buffer[index2] = resistance;
    index2++;
    if (index2 >= NUM_READS) index2 = 0;
}

...

// Cálculo del promedio de las lecturas sobre cada sensor

long average() {
    long sum = 0;
    for (int i = 0; i < NUM_READS; i++) {
        sum += buffer[i];
    }
    return (long) (sum / NUM_READS);
}

...

```

También existe un sensor de temperatura y humedad en cada plataforma de sensores, siendo los más populares, en cuanto a la temperatura, los de la familia DS18B20 del fabricante *Dallas*

Temperature. A continuación se describe el código que ejemplariza cómo conseguir la temperatura y guardarla en la variable *temp*.

```
#include <OneWire.h>
#include <DallasTemperature.h>

//set up temp sensor
#define ONE_WIRE_BUS 12
OneWire ourWire(ONE_WIRE_BUS);
DallasTemperature sensors(&ourWire);

//Read DS18B20 sensor
float temp;
sensors.requestTemperatures(); // Comando para obtener la temperatura
//temp = (sensors.getTempFByIndex(0)); // Grados F
temp = (sensors.getTempCByIndex(0)); // Grados C
```

Actuadores asociados

- Apertura y cierre del riego del área monitorizada por la plataforma de sensores y actuadores.

En este caso, no se enviará a la *blockchain* ningún dato sobre la actividad de los actuadores. Su comportamiento se conocerá por la información aportada por los sensores de humedad.

LM-210 radio module. Es el módulo encargado de modular y de emitir los datos desde el Vinduino al *gateway*. El envío de las lecturas de los sensores se realizará cada 15 minutos, que es cuando pasa a estar activa la plataforma de sensores. Además, dentro del ciclo de lectura de sensores y envío de las lecturas se añade un retardo seleccionado al azar, así que es prácticamente imposible que se produzcan colisiones en la red LoRaLAN del viñedo.

Los parámetros de la comunicación se han configurado para comunicarse con el *gateway* Raspberry Pi con protocolo asíncrono start/stop y a 9600bits por segundo, iniciándose el envío de las tramas de información con el carácter «\$» y finalizándolo con el carácter «!». El código para enviar la información es:

```
char loraMac [ ] = "xxxxxx"; // Identificador único de la plataforma

// Inicialización de las comunicaciones a 9600 bps
Serial.begin(9600); // Inicialización del módulo LoRa

// setting up the LM-210 LoRa module control pins:
pinMode(3, OUTPUT); // módulo LoRa, pin P1
pinMode(4, OUTPUT); // módulo LoRa, pin P2

// P1=0 and P2=0 : module active mode (Mode 1)
// P1=0 and P2=1 : module wake-up mode (Mode 2)
// P1=1 and P2=0 : module power saving mode (Mode 3)
// P1=1 and P2=1 : module set-up mode (Mode 4)

.....
```

```

digitalWrite(4, LOW);      // módulo LoRa, pin P2
digitalWrite(3, LOW);      // módulo LoRa, pin P1
digitalWrite(13, HIGH);    // LED ON

if (Serial.available() > 0)
{
int randomNumber=random(5000, 10000);
delay (randomNumber);}

// Imprimir / enviar los resultados
Serial.print("$");
Serial.print(",");
Serial.print(loraMac);
Serial.print(",");
Serial.print(read1);
Serial.print(",");
Serial.print(read2);
Serial.print(",");
Serial.print(read3);
Serial.print(",");
Serial.print(read4);
Serial.print(",");
Serial.print(Vsys,2);
Serial.print(",");
Serial.print(temp);
Serial.print(",");
Serial.print(humidity);
Serial.println(",!");
delay (1000);

digitalWrite(13, LOW); // LED off

return;

}

```

Trama de información. Los datos enviados desde cualquier plataforma de sensores al *gateway* tienen la siguiente estructura en su *payload*:

```

write_key = "id"           # Identificador de la plataforma de sensores.
field_1 = nnnnnnnnnn      # Sensor a 2 metros de profundidad.
field_2 = nnnnnnnnnn      # Sensor a 1 metro de profundidad.
field_3 = nnnnnnnnnn      # Sensor a 0,5 metros de profundidad.
field_4 = nnnnnnnnnn      # Sensor a 0,05 metros de profundidad.
v_batt = v,v              # Voltaje de la batería.
temperature = tt,cc       # Temperatura en el entorno de la plataforma, °C.
humidity = hh             # Humedad en el entorno de la plataforma.
aux = "aux"               # Reservado para usos futuros.

```

Finalmente, el código que ha servido de base para configurar el IoT que permite el envío de la información capturada en las plataformas de sensores al *gateway* se encuentra en github¹⁵.

¹⁵https://github.com/ReiniervdL/Vinduino/blob/master/Vinduino-R3/Vinduino-R3_DS18B20/Vinduino-R3_DS18B20.ino

5.1.3 Gateway Raspberry PI

El servidor RaspBerry Pi, en todo momento denominado en este TFM como el *gateway* del viñedo, es donde se han realizado actualizaciones para que la información captada por la plataforma de sensores del viñedo sea enviada a la *blockchain*.

A continuación se detallan las características y tareas principales relacionadas con el *gateway*, la mayoría de ellas ejecutadas con el software agnóstico Python 3.x.

- Sensores asociados al *gateway* del viñedo (estación meteorológica)
 - Velocidad del viento
 - Dirección del viento
 - Pluviometría
 - Sensor humedad del ambiente
- Comunicaciones
 - Radio LoRa
 - Se está usando un adaptador LD20-H USB LoRa del fabricante GlobalSat. Estadísticamente, las redes LoRaLAN soportan hasta 300 plataformas de sensores enviando información sin riesgo de colisiones.
 - Recepción de datos desde las plataformas Vinduino. Protocolo asíncrono start/stop, 9600bits.
- Wifi Internet
 - Protocolo TCP/IP
 - Envío de los datos vía Internet a:
 - Thingspeak
 - *Blockchain* (almacenamiento de los datos)
- Almacenamiento de la información
 - Todos los datos se almacenarán, antes de su envío, en un disco local montado en la RaspBerry PI para poder compensar posibles pérdidas del enlace TCP/IP y así realizar envíos posteriores.

Piezas de software en el *gateway*

1. **Comunicaciones LoRaLAN.** Los vinduinos del viñedo reportan las lecturas de sus sensores asociados en modo *broad-cast* y estas emisiones son recibidas por el módulo LoRa ubicado en el Vinduino que está en modo escucha. Con las siguientes líneas de código se logra interpretar el mensaje recibido y almacenarlo en variables para su tratamiento posterior, ya sea para almacenar los datos en la base de datos local por razones de seguridad, o bien para su envío a Thingspeak. En este trabajo de fin de

master se han introducido líneas de código para gestionar el envío de información a la *blockchain*.

El código de recepción de la información en el *gateway* es:

```
from time import localtime, strftime
import serial

ser = serial.Serial('/dev/cu.usbmodem1411', 9600, timeout=900)
while True:
    line=ser.readline()
    if len(line)==0:
        print("Time Out")
        sys.exit()
    line=line.decode("utf-8")
    print (line)
    print (len(line.split(",")))

    try:

        start_char = line.split(",")[0]
        write_key = line.split(",")[1]
        field_1 = line.split(",")[2]
        field_2 = line.split(",")[3]
        field_3 = line.split(",")[4]
        field_4 = line.split(",")[5]
        v_batt = (line.split(",")[6])
        temperature = line.split(",")[7]
        humidity = line.split(",")[8]
        aux = line.split(",")[9]
        stop_char = line.split(",")[10]

    except:
        if start_char != "$":
            print ("no start character detected")
```

2. Lectura de los datos de la estación meteorológica ligada al BUS de comunicaciones I2C de la Raspberry Pi.

- Procedimiento de lectura y envío a la *blockchain* aplazado para publicaciones futuras.

3. Almacenamiento de la información recibida de las plataformas de sensores Vinduinos en una base de datos SQLite, para solucionar problemas temporales de acceso a Internet.

```
import sqlite3
from time import localtime, strftime

conn = sqlite3.connect('vinduino.db')
print ("Opened database successfully",)
conn.execute('INSERT INTO VINDUINO (TSKEY, STATION_ID, SENSOR1,SENSOR2,
SENSOR3, SENSOR4, VBAT, TEMP, HUMID, AUX) \ VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?,
?,?)', ((strftime("%d %b %Y %H:%M:%S", localtime()), write_key, field_1,
field_2, field_3, field_4, v_batt, temperature, humidity, aux))
conn.commit()
```

```
print ("Records created successfully",)
conn.close()
```

4. Reenvío de la información a *Thingspeak* para su tratamiento y construcción de gráficas en tiempo real.

La Tabla 1 muestra la relación entre **id** del Vinduino y los canales habilitados en *Thingspeak*. Estos valores se relacionan en la variable `write_key`.

Vinduino id.	write_key	Channel URL
Vinduino1_PV (Averiado 23/4)	V55XYQX29JTFWNVR	https://thingspeak.com/channels/310994
Vinduino2_CS	2MFWMJNVGMWMEFO	https://thingspeak.com/channels/311015
Vinduino3_MBE	IORG1AH0DC1B56LL	https://thingspeak.com/channels/311018
Vinduino4_MBW	XBA5TPH2RV65Q52K	https://thingspeak.com/channels/311022

Tabla 1. Correspondencia entre el ID de la plataforma y el canal en ThingSpeak

5. Envío de los datos a la *blockchain*, que será la base de datos permanente.

```
import requests
import json
from time import localtime, strftime

tskey = (strftime("%a, %d %b %Y %H:%M:%S", localtime()))

user_id = "raspberry_pi"
password = "palabra de paso"
access_token = "Lo exigido por blockchain Side-Chain"

url = 'https://httpbin.org/post'
headers = {'Content-Type': 'application/json',}
params = {'access_token': access_token,}
payload= json.dumps({
    'recipient':{'id':user_id, 'passwd': password,},
    'campo1':tskey,
    'campo2':write_key,
    'campo3':field_1,
    'campo4':field_2,
    'campo5':field_3,
    'campo6':field_4,
    'campo7':v_batt,
    'campo8':temperature,
    'campo9':humidity,
    'campo10':aux
})
r = requests.post(url, headers=headers, params=params, data=payload)
```

```
print (r.status_code)
print (r.json())
```

Como se puede observar, aún hay parámetros pendientes de asignar, pues dependen de la *hand shake* que se establezca para escribir en la *blockchain*, pero este *webservice* se ha podido validar contra el sitio web: url = '<https://httpbin.org/post>', el cual responde en las pruebas con el código 200 y retorna la información enviada:

```
runfile('C:/Users/joaquin.jimenez/Dropbox/1_UNIR/TFM/webservices/webserviceVinduino
s.py', wdir='C:/Users/joaquin.jimenez/Dropbox/1_UNIR/TFM/webservices')
200
{'args': {'access_token': 'El que diga salva'}, 'data': '{"recipient": {"id":
"raspberrypi", "passwd": "palabra de paso"}, "campo1": "Sun, 10 Jun 2018
20:26:35", "campo2": "Vinduino1_PV", "campo3": 4294967220, "campo4": 4294967220,
"campo5": 4294967220, "campo6": 4294967220, "campo7": 4.5, "campo8": 3, "campo9":
72, "campo10": "aux"}', 'files': {}, 'form': {}, 'headers': {'Accept': '*/*',
'Accept-Encoding': 'gzip, deflate', 'Connection': 'close', 'Content-Length': '279',
'Content-Type': 'application/json', 'Host': 'httpbin.org', 'User-Agent': 'python-
requests/2.18.4'}, 'json': {'campo1': 'Sun, 10 Jun 2018 20:26:35', 'campo10':
'aux', 'campo2': 'Vinduino1_PV', 'campo3': 4294967220, 'campo4': 4294967220,
'campo5': 4294967220, 'campo6': 4294967220, 'campo7': 4.5, 'campo8': 3, 'campo9':
72, 'recipient': {'id': 'raspberrypi', 'passwd': 'palabra de paso'}}, 'origin':
'85.58.88.118', 'url': 'https://httpbin.org/post?access_token=El+que+diga+salva'}
```

5.1.4 Trabajadores

- Sensores manuales
 - Grado de azúcar de la uva
 - Tareas realizadas en el viñedo
- Comunicaciones
 - Apps móviles reportando los datos tomados manualmente vía formularios web (HTTP, POST) a la *blockchain*

5.1.5 Clientes y distribuidores

Los clientes o usuarios finales también serán una fuente de información, cuando estos envían a la *blockchain* su experiencia al consumir el vino. Los consumidores accederán de forma controlada al servicio de almacenamiento de datos en la *blockchain*, donde podrán describir su experiencia. El identificador para la app del usuario que permitirá el acceso a la *blockchain* constará de dos partes: código QR leído en la botella y un identificador que encontrará en el tapón de la botella. En el momento de depositarse la información en la *blockchain*, con la ayuda de oráculos podrán correlacionarse informaciones de interés acerca de la historia de la botella en cuestión.

Otra fuente de información en el último tramo del ciclo del vino pueden ser los distribuidores, quienes con sus dispositivos móviles dotados de NFC podrán leer la información de sensores actualmente adosados a los pallets o cajas de transporte e informar de forma automática si la

botella ha estado sometida a altas temperatura, traqueteos, luz solar, etc. La información recogida podrá enviarse a la *blockchain* utilizando una app diseñada a tal efecto. En este caso también se hará uso de los «oráculos», que en el momento en que el distribuidor envíe los datos captados en el pallet —antes de ser almacenados en la *blockchain*— complementarán la información, indicando qué botellas de los lotes controlados han sido afectadas, etc.

5.2 Repositorio de la información

En este apartado se expondrá cómo la información enviada por el *gateway* (hoy instalado en el viñedo, mañana también en bodega, transporte, etc.) será almacenada de forma fiable e inalterable para siempre en la *blockchain*. El sitio web asociado mostrará alguna información en tiempo real y podrán hacerse consultas posteriores sobre la historia del vino y estudios específicos de forma forense.

Todos los registros de información almacenados estarán identificados por una clave (*Time Stamp Key*) estampada en el momento en que los datos son reenviados desde el *gateway* del viñedo a la base de datos o son enviados por trabajadores, distribuidores o consumidores. Las agrupaciones de datos que compongan una entidad de información serán firmadas adecuadamente y almacenadas en una cadena de bloques. Esta cadena de bloques ha de garantizar la integridad de la información, certificará el no repudio y asegurará la existencia de los datos ilimitadamente en el tiempo.

5.2.1 La Cadena de bloques

We can't yet predict what the blue-chip industries built on blockchain technology will be, but we are confident that they will exist, because the technology itself is all about creating one priceless asset: trust.
(Casey, 2018)

La propuesta de almacenamiento de la información, que es recogida por los diferentes sensores dispersos por el viñedo y los sensores ubicados en la estación meteorológica y enviada desde los dispositivos móviles por los trabajadores, los distribuidores o los clientes para reflejar su experiencia con el vino, es almacenarla en un tipo de *blockchain* conocido como *side chain*. Esta será pública y se regirá por unas reglas de permisos (*permissioned*) que determinarán quién puede leer o escribir en la cadena de bloques, así como quién puede ejecutar determinadas órdenes.

Dentro del ecosistema de cadenas de bloques, se puede situar la *side chain* (que se describirá más adelante) en la siguiente tabla definida por McKinsey. La *side chain* propuesta corresponde a cadenas de bloques públicas y con gestión de permisos.

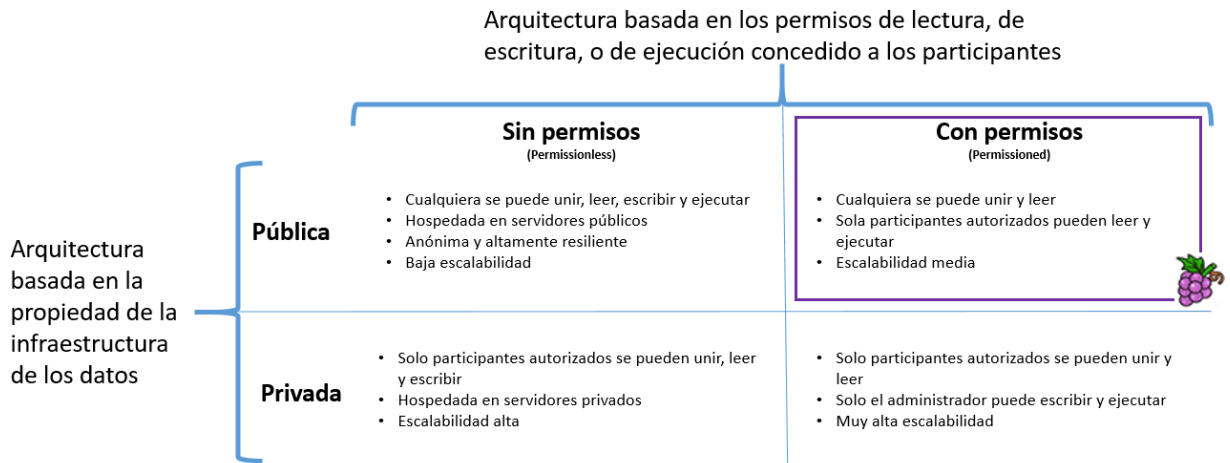


Tabla 2. Tipos de blockchain. Fuente: McKinsey&Company

La *side chain* proporciona el almacenamiento de los datos recolectados asegurando su integridad junto con su validación. Esta propuesta responde a la necesidad de dotar de confianza y asegurar la verdad de todos los datos almacenados, que posteriormente se utilizarán para construir la trazabilidad del vino. La propuesta de *side chain* elaborada con Jordi Estapé y Mike Barrow (Estapé, 2018) dentro del proyecto Openvino reduce enormemente los costes asociados al almacenamiento de los datos en su totalidad en una *blockchain* pública, como pudiera ser Ethereum (Dannen, 2017).

Por lo tanto, se puede concluir que definitivamente la aportación de una *side chain* incrementa el valor de la trazabilidad, en cuanto que:

- **Incrementa la confianza.** Se incorpora la figura de los validadores. Estos serán responsables de certificar los datos recolectados y asegurarán su veracidad.
- **Inmutabilidad.** La solución de *side chain* propuesta implica la implementación de un árbol de *merkle* (algoritmo criptográfico) que asegurará la inmutabilidad de los datos validados.
- **Reducción de costes.** Los costes asociados a una transacción en esta *side chain* son económicamente muy baratos y, en cuanto a la ejecución, más rápidos, en comparación con cualquier transacción que se realice sobre una cadena pública tipificada como *permissionless (main chain)*. Por lo tanto, esta propuesta conduce a una reducción de costes y muestra una única verdad en la trazabilidad del vino.

5.2.2 Propuesta de almacenamiento en la *blockchain*

La propuesta para la capa de almacenamiento, por lo tanto, se desdoblará en dos cadenas de bloques, y ambas serán cadenas públicas. Una cadena es la *side chain*, gestionada y regulada por OPENVINO con reglas de acceso para su escritura, y la otra, la principal, aún

pendiente de definir, será totalmente pública, no presentará ninguna restricción de permisos y estará regulada por la desconfianza de sus miembros.

Desde este momento, los artefactos y actores de la capa física vista en apartados anteriores (y que en este trabajo está compuesta por las plataformas de sensores, la estación meteorológica, los trabajadores del viñedo, los distribuidores y los consumidores) serán quienes alimenten la capa de almacenamiento. En este nivel aparece un nuevo actor, que es el propietario y administrador de la *side chain*. Sin embargo, esto no significa que sea otra fuente de datos para la capa de almacenamiento.

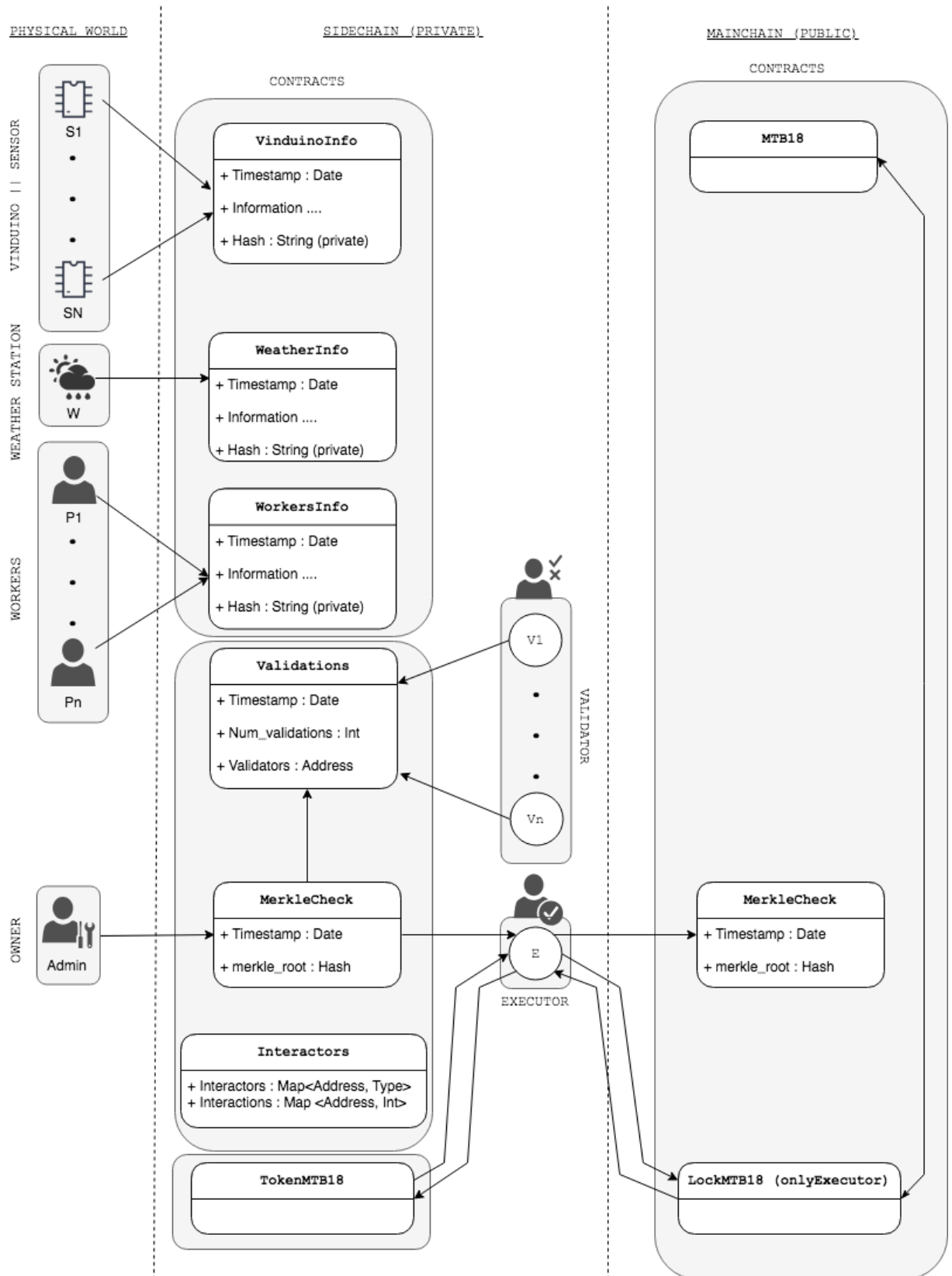


Figura 13. Almacenamiento de la información. Fuente: Jordi Estapé

Antes de definir las de manera exhaustiva, se puede decir que las dos cadenas de bloques que formarán la capa de almacenamiento, detalladas en la figura anterior, aportan:

1. **Side chain.** Esta cadena, pública con ciertos privilegios de acceso, además de almacenar todos los datos generados en la capa física (viñedo y otras fuentes de datos) en forma de contratos inteligentes, también incorpora dos entidades relacionadas: el ejecutor y el validador.
2. **Main chain.** Esta cadena, pública sin privilegios, aporta seguridad y confidencialidad total. En esta cadena se mantienen identificados los *tokens* y los registros de validación (*hashes*), que determinan que la *side chain* no ha sido alterada.

5.2.3 Capa física de aporte de datos a la *side chain*

Esta capa tiene la responsabilidad de enviar información a la cadena de bloques. Todas las entidades de esta capa, menos el «administrador u *owner*», deben ser tratadas como fuentes de datos.

- **Vinduiños.** Los Vinduiños deben enviar la información recolectada a la *side chain* vía la puerta de salida (Raspberry Pi). Cada Vinduiño, con una MAC única, la del módulo LoRa, deberá tener asignada una pareja de claves —pública y privada— desde la *side chain*, a fin de realizar transacciones que permitan su identificación para que los contratos puedan almacenar correctamente su información.
- **Estación meteorológica.** La estación meteorológica, ligada a la puerta de salida vía un bus I2C, debe enviar los datos a la *side chain* con una determinada frecuencia, que podría ser similar a la frecuencia de los vinduiños, es decir, cada 15 minutos. La estación meteorológica también debe poseer las dos claves (pública y privada) de la *side chain* para que su información sea gestionada por el contrato pertinente de la *side chain*.
- **Workers.** Los trabajadores del viñedo deben reportar el resultado de sus tareas a la *side chain* vía un software (no se determina aún), el cual debe tener la capacidad de gestionar un monedero (*wallet*), para así transaccionar con la *side chain*. El uso del *wallet* almacenado en el dispositivo móvil permitirá identificar a los trabajadores en la transacción.
- **Consumidores.** Otra fuente de datos serán los clientes o consumidores. Con una app diseñada a tal efecto, podrán leer con la cámara del teléfono móvil las etiquetas QR, y con una contraseña de acceso que obtendrán, por ejemplo, en los tapones de botellas, podrán obtener los permisos de escritura en la *side chain*, vía los procedimientos que se establezcan.
- **Distribuidores o tiendas de vino.** Con tecnología NFC acopladas a los teléfonos móviles, estos actores podrán capturar la información de sensores RFID ubicados en pallets o cajas de botellas y trasladarla a la *side chain*. Un código complementario para

obtener la capacidad de escribir en la *side chain* podría ser el id de la transacción por la cual adquirieron los *tokens winecoin* MTB.

- **Owner.** Este usuario especial tiene la responsabilidad de administrar la cadena *side chain* en lo que respecta a:
 - A. Seleccionar los validadores (deberá existir un método de autoselección para evitar deficiencias en este proceso).
 - B. Ejecutar el algoritmo «*merkle root*», así como sus copias en la *blockchain* principal.
 - C. Ejecutar las transferencias de los *tokens* MTB18 de la *side chain* a la *blockchain* principal.
 - *) Tanto A como B serán detallados más adelante dentro de los apartados *Side chain* y *Main chain*.

5.2.4 *Side chain* (pública con privilegios)

Esta cadena es la responsable de validar y de almacenar los datos, a la par que ofrece un coste reducido por transacción.

- **Validadores.** Los validadores son personas que tienen la responsabilidad de validar que los datos que se almacenan son correctos. Estos validadores no han de tener relación con el viñedo, pueden ser sencillamente clientes o consumidores que se han ganado el derecho a ser validadores. Los validadores trabajarán en el proceso de validación porque obtendrán una recompensa por realizar este trabajo (similar a lo que ocurre con los mineros en la *blockchain* del *bitcoin*). El precio todavía se tiene que definir, pero puede ser una cantidad de *winecoins* MTB18* proporcional a la cantidad de trabajo realizado. Por esta razón, los contratos de los validadores tienen un contador de validaciones por cada validador.
 - La captación de validadores (mineros) se puede producir en el momento en que un consumidor tiene a su alcance una botella MTB y lee el código QR en la parte trasera de la botella. El código QR lo llevará al sitio web que mostrará la trazabilidad del vino, y lo invitará a ganar el derecho a ser validador si realiza las acciones siguientes:
 - Descargar la app relacionada con este proyecto
 - Habilitar la ubicación en el dispositivo móvil para su envío a la *side chain*
 - Introducir el código impreso en el tapón de la botella
 - Habilitar que se envíe la información captada en el código QR de la botella

- Una vez adquirido el derecho a ser validador, se le asignará un código para que pueda actuar como tal. El sistema entregará un código validador único por etiqueta QR, más el código del tapón de la botella, y se le asignará una duración determinada para actuar como tal. Lecturas posteriores de QR más contraseñas diferentes podrán alargar la concesión de validador. Se determinará cómo gestionar la identificación del validador, pues, como se verá más adelante, recibirá una remuneración por su trabajo
- Existirá un validador especial y que tendrá un incentivo extra por enviar la información que pueda capturar con su dispositivo NFC, al leer los posibles sensores RFID adosados en pallets o cajas de botellas, para de esta manera reportar la trazabilidad del desplazamiento de la botella.
- **Ejecutor.** El ejecutor tiene la responsabilidad de transferir datos desde la *side chain* a la *blockchain* principal, así como de ejecutar las actualizaciones del algoritmo *merkle root*.
 - A. **Contratos.** Hay tres subgrupos de contratos inteligentes¹⁶.
 - B. **Data storage.** Se utiliza este contrato inteligente para almacenar información. Todos los contratos tienen una estructura similar, que contiene: un *timestamp* (tiempo de la transacción), información (datos recolectados) y un *hash* generado a partir de la concatenación del *timestamp* y de la información.
 - C. **Validity and immutability.** Este paquete está compuesto de tres contratos diferentes:
 - **Interactors.** Contienen una descripción de cada tipo de usuario válido. Este contrato permite controlar los permisos sobre la *side chain* y también da la posibilidad de ofrecer a los usuarios una breve descripción y el identificador del «*Interactor*».
 - **Validation.** Su contenido está compuesto de un *timestamp*, un contador de validaciones y un listado con las direcciones de quienes han validado la transacción. Este contrato controla la validación de los datos vía auditorías externas conocidas como «*validators*». Se supone que cada *x* tiempo (tiempo por determinar) los *validators* validan que los datos recolectados son correctos e interactúan con este contrato para indicar que todo está funcionando correctamente. Una validación de un

• ¹⁶ Importante: esto es solo una idea abstracta, no es la solución completa. Los contratos no se especifican en detalle, solo se hace referencia a su estructura.

timestamp dado, valida instantáneamente todas las entradas de validaciones previas. Por lo tanto, si V1 valida a las 13:59 y V2 valida a las 18:00 horas, la validación de las 13:59 habrá sido realizada por V1 y V2. Sin embargo, la validación de las 18:00 horas habrá sido realizada solo por V2. Se consideran datos válidos (datos almacenados en los contratos) aquellos cuyo *timestamp* sea inferior al último *timestamp* validado por más del 50% de los validadores (este porcentaje es configurable).

- **Merkle Tree.** Cada x tiempo, el ejecutor ejecutará este contrato para calcular un *merkle tree* con los siguientes valores:
 - El *merkle root* previo (1 hoja).
 - La nueva información añadida y validada (n hojas).

Esto retornará un *hash* total que representará el estado real de la información. Si se cambia algún dato histórico y se recalcula el *merkle root*, el *hash* será diferente y se detectará la anomalía.

- D. Criptomoneda MTB18. La *side chain* eliminará un efecto colateral en las transacciones de los *winecoins* MTB* que se explicará en el apartado de la *main chain*.

5.2.5 Main chain (cadena principal y pública)

En este TFM la cadena principal hace uso de la *blockchain* pública Ethereum. El coste de las transacciones de esta cadena principal es más alto que el de las transacciones en la cadena *side chain*, donde el coste es despreciable en comparación con el de las públicas, tanto en el coste pagado a los mineros como en el computacional. Como consecuencia de ello, solo quedará almacenada en la *blockchain* Ethereum una copia del histórico del *Merkle Root* de la *side chain*, y esto será suficiente para incrementar la inmutabilidad. Sin embargo, la *blockchain* Ethereum es la cadena en la que los valores del *token* presentan la confianza y seguridad máximas.

Por lo tanto, aquí se abre otra vía de acción, consistente en hacer intercambios de criptomonedas MTB* con el soporte de la *side chain*. Una vez que determinados *tokens* sean intercambiados y validados en la *side chain*, pasarán a estar almacenados en la *blockchain* Ethereum, para proteger el sistema en caso de fallo en la *side chain*.

- **MTB18.** El *winecoin* MTB18 implementado y creado en el momento en que Costaflores convirtió en vino su cosecha del 2018. Una botella de vino MTB del 2018 es igual a un *token* MTB18.

- **Merkle check.** Copia histórica de los datos del contrato MerkleTreeCheck localizado en la *side chain*.
- **LockMTB18.** Este es un contrato que bloquea los *tokens* MTB18 para transferirlos a la *side chain*. Cuando el usuario quiera que los *tokens* vuelvan a la cadena principal, solo tiene que desbloquearlos quemándolos en la *side chain*.

Ejemplo de gestión de los *tokens* MTB18.

Supongamos que cuatro amigos, F1, F2, F3 y F4, han comprado 6 *tokens* MTB18 cada uno y se realizan las siguientes acciones:

F1 bloquea 6 *tokens* MTB18

(CP) MTB18	(CP)LOCKMTB18	(SC)MTB18
F1=0, F2=6, F3=6, F4=6	MTBLock = 6	F1=6, F2=0, F3=0, F4=0

F1 transfiere 3 MTB18 a F2

(CP) MTB18	(CP)LOCKMTB18	(SC)MTB18
F1=0, F2=6, F3=6, F4=6	MTBLock = 6	F1=0, F2=3, F3=0, F4=0

F1 transfiere 2 MTB18 a F3

(CP) MTB18	(CP)LOCKMTB18	(SC)MTB18
F1=0, F2=6, F3=6, F4=6	MTBLock = 6	F1=0, F2=3, F3=2, F4=0

F1 transfiere 1 MTB18 a F4

(CP) MTB18	(CP)LOCKMTB18	(SC)MTB18
F1=0, F2=6, F3=6, F4=6	MTBLock = 6	F1=0, F2=3, F3=2, F4=1

F4 transfiere 1 MTB18 a F3

(CP) MTB18	(CP)LOCKMTB18	(SC)MTB18
F1=0, F2=6, F3=6, F4=6	MTBLock = 6	F1=0, F2=3, F3=3, F4=0

F2 transfiere 3 MTB18 a F3

(CP) MTB18	(CP)LOCKMTB18	(SC)MTB18
F1=0, F2=6, F3=6, F4=6	MTBLock = 6	F1=0, F2=0, F3=6, F4=0

F3 desbloquea 6 MTB18

(CP) MTB18	(CP)LOCKMTB18	(SC)MTB18
F1=0, F2=6, F3=12, F4=6	MTBLock = 0	F1=0, F2=0, F3=0, F4=0

Tabla 3. Transacciones de tokens en la side chain

La conclusión de las transferencias expuestas anteriormente muestra que se han realizado transferencias en la *side chain* sin coste alguno o despreciable, y ello ha sido como consecuencia de aparecer una sinergia entre la trazabilidad del vino aquí desarrollada y el intercambio de criptomonedas. Sin embargo, lo que parece un aporte tecnológico que abarata transacciones puede estar en conflicto con las leyes fiscales del país donde sean residentes estos amigos, que se han tomado aquí como un caso de uso.

5.2.6 Scripts de soporte temporales

Durante este TFM se ha considerado que, para una puesta en marcha inicial, toda información llegada desde la capa física a la *side chain*, ya sea información de las plataformas de sensores, de la estación meteorológica o de usuarios móviles, será recogida por unas API que respondan a determinados mensajes del tipo POST especializados para cada caso. Así que, en el mismo bucle de recepción de los datos, estas API tomarán la información del mensaje y la depositarán en la *side chain*.

Serán estos *scripts* los encargados de gestionar las parejas de claves, pública y privada, asociadas a cada origen para poder acceder a los contratos de la *side chain* y así depositar la información que vayan recibiendo de la capa física. A la par, estas API podrán interactuar con «oráculos» para obtener informaciones complementarias. Trasladar la gestión de las claves privadas al origen de los datos será un objetivo de próximos trabajos.

5.3 WEB

En el actual sitio web de Costaflores se habilitará un espacio para mostrar o consultar la trazabilidad del vino, al que se accederá desde el QR de la botella MTB y ofrecerá al público los detalles de la añada en el viñedo que dio lugar al vino en la botella. En el futuro se ampliará la información con los datos de la bodega y más tarde con los datos que recojan los sensores en cajas o pallets que transportan unas botellas determinadas en sus viajes por el mundo.

La información que se mostrará en formato web o, en un futuro, en apps realizadas con fines determinados será recopilada de la *side chain* y puesta a disposición de quienes la requieran, ya que en la *side chain* los derechos de lectura serán universales.

5.3.1 Historia de temperatura en el viñedo

En este caso de uso, una primera propuesta para mostrar la temperatura en el viñedo sería disponer de un software *proxy*, el cual tomará todos los datos requeridos de la *side chain* y los preparará para que los explote quien realice las consultas. El resultado de las lecturas de las temperaturas en la *side chain* y su procesamiento aportará una ristra de información en formato json (parejas *código:valor*), que indicará, por ejemplo, la temperatura máxima y mínima de cada día en el viñedo.

Ejemplo: un_mes_de_verano.json

```
{ "year": 2018, "month": 1, "temphigh": [ 33, 35, 34, 30, 30, 35, 36, 33, 36, 32, 35,
37, 33, 36, 33, 36, 37, 34, 39, 39, 39, 37, 34, 31, 33, 32, 37, 41, 41, 37, 35 ],
"templow": [ 21, 19, 22, 19, 19, 19, 21, 20, 22, 21, 21, 24, 25, 20, 22, 16, 22, 24,
25, 27, 25, 21, 22, 23, 20, 19, 19, 20, 20, 24, 20 ] }
```

Para mostrar temperaturas, una opción es optar por tecnologías SVG, y la buena información para soportar esta estrategia de representación se consigue con las teorías explicadas en «Path Data» (w3org, 2018).

5.3.2 Lluvia y radiación solar

De igual manera se puede obtener de la *side chain* las precipitaciones y radiaciones solares máximas de cada día. A continuación se muestran ficheros json con datos de un supuesto día de verano.

```
{ "year": 2018, "month": 1, "rainfall": [ 0, 0, 0, 0, 0, 0, 1, 0, 6, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] }
```

```
{ "year": 2018, "month": 1, "solar": [ 5.9, 4.8, 5.1, 5.4, 5.8, 5.2, 5.2, 4.6, 5.1,
5.6, 4.7, 5, 4.4, 4.1, 4.7, 5.6, 4.3, 4, 4.2, 5.6, 4.1, 5, 5.8, 4, 4.9, 4.6, 5.5,
4.2, 5.8, 4, 5.9 ] }
```

5.4.2 Tareas en el viñedo

Después de que los trabajadores informen sobre las tareas en el viñedo, estas se han de discutir con expertos del negocio para determinar qué importancia tienen y cómo

correlacionarlas en la historia de un vino. Informáticamente es fácil de reflejar: se puede obtener el fichero json de la *side chain* y mostrarlo posteriormente en una gráfica asignando iconos que sean de fácil reconocimiento.

Las tareas de un mes en el viñedo podrían ser:

```
{ "year": 2018, "month": 1, "tasks": [ { "day": 1, "id": "icon-white-pala",
"text": "Escarbar 30 minutos" }, { "day": 2, "id": "icon-white-martillo", "text":
"Reparaciones 30 minutos" }, { "day": 3, "id": "icon-white-pala", "text": "Escavar
la viña 60 minutos" }, { "day": 7, "id": "icon-white-personas", "text": "Personas
60 minutos" }, { "day": 9, "id": "icon-white-caballo", "text": "Herradura 45 minutos"
}, { "day": 11, "id": "icon-white-tijera", "text": "Podar viña 45 minutos" }, {
"day": 13, "id": "icon-white-tractor", "text": "Arar 120 minutos" }, { "day": 14,
"id": "icon-white-caballo", "text": "Herradura 35 minutos" }, { "day": 17, "id":
"icon-white-martillo", "text": "Martillo 30 minutos" }, { "day": 19, "id": "icon-
white-pala", "text": "Herradura 75 minutos" }, { "day": 20, "id": "icon-white-
caballo", "text": "Herradura 15 minutos" }, { "day": 21, "id": "icon-white-caballo",
"text": "Herradura 30 minutos" }, { "day": 24, "id": "icon-white-caballo",
"text": "Herradura 45 minutos" }, { "day": 25, "id": "icon-white-caballo", "text":
"Herradura 20 minutos" }, { "day": 27, "id": "icon-white-caballo", "text": "Herradura
20 minutos" }, { "day": 28, "id": "icon-white-martillo", "text": "Reparaciones 30
minutos" }, { "day": 31, "id": "icon-white-vacaciones", "text": "Vacaciones 60
minutos" } ] }
```

5.4.3 Propuestas de representaciones gráficas responsivas

Según mis tutores en este aspecto, y tal como se refleja en la wiki (González & Gea, 2018), los apartados anteriores deberían mostrar una gráfica-resumen con esta forma:



Figura 14. Gráfica de tareas y condiciones atmosféricas de un mes

Nota: La gráfica y los datos anteriores no están correlacionados, se ofrecen aquí a modo de ejemplo.

Por otro lado, si el fichero *json* hiciera referencia a unos datos cualesquiera de un año, un formato básico de presentación sería, por ejemplo:



Figura 15. Gráfica circular con datos tabulados para un año

Para construir la gráfica anterior, según exigencias SVG del w3.org, se deberán transformar los datos originales en un paso intermedio como un «string path»¹⁷, que para los primeros cuatro meses sería:

```
sPathTempMax = M330,330 10,-320 L488,56 L597,176 L629,330 L580,474 L470,572 L330,608
L187,578 L77,476 L27,330 L60,174 L171,55 L330,10
```

5.4 Historia de un vino MTB*

Accediendo a la historia de una botella MTB*, antes de entrar en los detalles de la trazabilidad, se presentará al consumidor un relato, vía web o vía app, confeccionado con información que afectará al vino de la botella en cuestión. El relato hará referencia a todo el ciclo del vino de la botella, comenzando por características del viñedo como son las cepas que han dado origen al vino de esa cosecha, la edad de las cepas, la localización del viñedo, con sus coordenadas terrestres y su altura con respecto al nivel del mar, etc.

A continuación, se mostrarán en forma de relato navegable todos los detalles de la trazabilidad, y el lector podrá saltar, por ejemplo, a las opiniones de los consumidores de la cosecha a la que pertenece la botella en manos del consumidor.

La historia del vino en una botella MTB* hará una mención especial a la Responsabilidad Civil Corporativa (RSC) de Costaflores, esto es, se mostrarán testimonios de responsabilidad con los trabajadores en el viñedo, del compromiso con el medio ambiente, así como del cumplimiento de todas las obligaciones legales con el país de origen del vino, en este caso

¹⁷ <https://www.w3.org/TR/SVG/paths.html#PathData>

Argentina. Esta RSC deberá ser registrada en todos sus detalles en la *side chain* y certificada por los validadores.



Figura 16. Reflejo de un comercio justo

Un entorno de criptomonedas, donde el precio del vino lo ponen los consumidores, una trazabilidad certificada por una cadena de bloques, que en el caso de la *side chain* también es respetuosa con el medio ambiente, y una política de responsabilidad civil corporativa exigente son vectores de acción que, junto al paradigma OPENVINO, determinan un comercio justo alrededor del vino MTB.

6. Desarrollo de un prototipo

A continuación se describe cómo debería ser el prototipo con la finalidad de reproducir la arquitectura informática (hardware y software) definida en este TFM. De esta manera será posible diseñar, implementar y probar en el prototipo todos los puntos relacionados con la trazabilidad para los cuales aún no se ha ofrecido una solución.

En la Figura 17 se describen los bloques mínimos funcionales para el funcionamiento del prototipo.

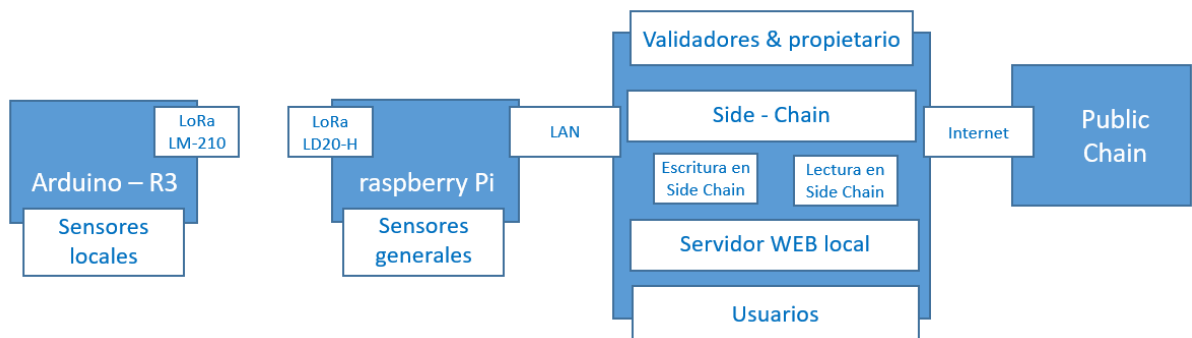


Figura 17. Maqueta para construir

El entorno descrito en la figura 17 podría considerarse un entorno de pre-producción. A continuación se describen los componentes que formarán la maqueta.

6.1 Plataformas de sensores

Las plataformas de sensores están basadas en hardware y software libre y los componentes para el prototipo han de ser un conjunto mínimo de elementos que permita simular la realidad de la sensórica que se ha de gestionar.

6.1.1 Plataforma en viñedos

El hardware recomendado y mínimo para construir la reproducción de las plataformas de sensores en viñedo es:

- Plataforma de sensores remotos Vinduino R3
 - Basada en el Arduino Pro Mini, con procesador ATmega328, 3.3V 8Hz
- Globalsat LM-210 LoRaLAN
- Watermark SS200 x 4
- DS18B20 sensor de temperatura

Se ha de disponer del cable FDTI para tareas de programación de los Vinduinos.

En el Anexo 1 se detallan los costos de adquisición de una plataforma de sensores.

6.1.2 Plataforma en bodega

Pendiente de definir

6.2 Gateways

6.2.1 Gateway en viñedo y estación meteorológica

El hardware recomendado para realizar el prototipo de la puerta de salida es:

- Raspberry Pi 3 Model B V1.2
 - SanDisk Extreme 64GB MicroSD
- The Weather Board Interface board
 - DS3231 Real time clock
 - Grove AM2315 - Encased I2C Temperature/Humidity Sensor
 - Conjunto de anemómetro, viento y lluvia

Y accesorios de conexionado.

El detalle de los precios para los conjuntos anteriores están en Anexo 3 y Anexo 2

6.2.2 Gateway en bodega

Pendiente de definir.

6.3 Dispositivos de los trabajadores

Los dispositivos de los trabajadores incorporarán una app que permita:

- a) Incorporar datos de los sensores manuales a la app, vía por ejemplo *bluetooth*
- b) Responder a cuestionarios sobre las tareas en el viñedo

6.4 Dispositivos de los usuarios

Los dispositivos de los usuarios podrán ser de dos tipos:

- a) Con solo lectores de códigos QR en los consumidores
- b) Con capacidad de leer códigos QR y leer sensores RFID en los distribuidores

La app de estos dispositivos podrá enviar datos a la *blockchain* como si fuera un elemento de la capa física y el formulario POST cumplimentado incorporaría:

- a) Código de la botella
- b) Datos de los sensores de la botella
- c) Opinión del usuario

El usuario, a cambio, obtendría un código único que le permitiría ser validador en la *side chain*.

6.5 Servidor

Para el prototipo se propone un servidor que albergue todas las piezas de software necesarias para dotar de trazabilidad al ciclo de producción del vino, desde el cuidado de las uvas hasta su consumo.

El servidor almacenará:

- Un servidor web, por ejemplo, LAMP (Linux, Apache, MariaDB, PHP...), que dará servicio a los usuarios en sus consultas.
- Unos servicios Proxy para elaborar la información requerida por los clientes web.
- Servicios de *script* para intermediar entre la capa física (viñedo, bodega, transporte y consumidores) y la *side chain*.
- La *side chain*
- Acceso de los validadores y propietario/ejecutor
- Pasarela a la *main chain*

6.6 Main chain (pública)

El acceso a una cadena pública implicará establecer el contrato de servicio con la cadena seleccionada. En este TFM se propone Ethereum.

7. Líneas de trabajo futuras

En un futuro, el trabajo de trazabilidad del viñedo definido en este TFM y los que se realicen cuando este se amplíe se irán incorporando al proyecto OPENVINO, que ha puesto sus recursos a disposición del desarrollo de este TFM. Estos recursos no solo se encuentran en el viñedo, sino también en la propia wiki OPENVINO, donde se están compartiendo opiniones con diferentes colaboradores o fuentes de conocimiento.

OPENVINO es un proyecto abierto en el que se exponen todos los detalles del vino producido en Costaflores, mostrando vía Internet a la comunidad de amantes del vino los métodos de trabajo, los costes de mantenimiento y elaboración del vino y sus balances en general. También se informa de que se trata de la primera entidad que ha sacado al mercado una moneda virtual respaldada por vino el 6 de mayo de 2018: los MTB18 *winecoins*.

Las líneas de trabajo futuras para completar la trazabilidad del ciclo de vida del vino consistirán en complementar la trazabilidad del viñedo expuesta en este TFM y se resumen a continuación.

7.1 Trazabilidad en bodega

La trazabilidad en bodega exigirá recopilar nuevos desarrollos de plataformas concentradoras de sensores basadas en Arduino, que deberán abarcar:

- **Elaboración del vino.** Este apartado debe contemplar el tipo de prensa, el porcentaje de extracción, la temperatura de fermentación, el tipo de depósito de fermentación, los tratamientos físicos durante el proceso, las posibles segundas fermentaciones (*maloláctica*), las condiciones y tipos de almacenaje, los sistemas de estabilización y clarificación, y los posibles *coupages*.
- **Envejecimiento del vino.** Tiempos y tipos de depósito de guarda, tipo y tiempo en barrica (crianza), sistemas de llenado, cambio y vaciado de las barricas.
- **Embotellado del vino.** Sistema de traslado del vino (por gravedad, mediante bombas), características del proceso de embotellado, tipo de tapón.
- **Envejecimiento en botella.** Condiciones de temperatura y humedad en la bodega.

7.2 Viaje de las botellas al consumidor

En este apartado se definirá cómo hacer el seguimiento de las botellas de forma individual. Están saliendo al mercado muchos tipos de sensores, cada día más baratos, que podrán hacer este seguimiento más fácil. Dicho seguimiento tendrá en cuenta los principales aspectos siguientes:

- **De la bodega a la estantería.** Los sensores han de registrar los cambios de presión y de temperatura a los que se somete el producto desde la salida de bodega hasta su destino final, traqueteos, exposiciones al sol, temperaturas máximas...
- **De la estantería al consumidor.** Se considerarán aspectos como la temperatura o la luminosidad.

7.3 Feedback del consumidor

Otro aspecto que se deberá trabajar en el futuro será el registro de las experiencias del consumo del vino de una determinada botella. Para que puedan registrarse en la *blockchain*, las opiniones de los consumidores no podrán ser anónimas.

7.4 Correlación de eventos

Otra propuesta de trabajo futuro es la aplicación de herramientas de *big data* para sustituir los servicios de monitorización en tiempo real que actualmente aporta Thingspeak, tal como se ha explicado anteriormente. Una posible herramienta de *big data* podría ser *ElasticSearch* (ELK). El envío de datos a Thingspeak podría ser sustituido por el envío de datos a una base de datos NoSQL, indexada por Lucene¹⁸, donde ELK pueda realizar actividades de correlación de la información o de descubrimiento de patrones y determinar, entre otras acciones, actividades predictivas dentro del ciclo del vino.

Se deberá determinar qué resultados de la monitorización con ELK y qué conclusiones de sus análisis deberán ir a la *side chain* para su almacenamiento permanente e integrarse en la trazabilidad del vino.

¹⁸ <https://github.com/fdv/elasticsearch-cluster-design-definitive-guide/blob/master/a-few-things-you-need-to-know-about-lucene.md>

8. Conclusiones

En este TFM se propone una solución tecnológica e innovadora que integra IoT y *blockchain* para realizar trazabilidades inalterables y fiables sobre cualquier producto.

La solución propuesta se compone de un conjunto de software y hardware con formato *GNU General Public License 3.0*, donde la arquitectura se ha estructurado en varias capas: la de captación de información, la de almacenamiento de la información en la *side chain* y la de mostrar la información. Este diseño permitirá evolucionar cualquier capa sin influir en el resto de ellas, y manteniendo inalterable la información almacenada. Esta arquitectura se está aplicando con éxito en el ámbito de la producción del vino y sus principios conceptuales podrán adaptarse a otros entornos, por ejemplo, en todos aquellos ámbitos en los que el IoT sea el elemento supervisor.

Por otro lado, se ha desarrollado una propuesta de captación de validadores, los cuales obtendrán *winecoins* como recompensa por su trabajo. Asimismo, durante la elaboración de este TFM se ha trabajado la posibilidad de sumar a la inmutabilidad y a la veracidad de la información de la *side chain* la alta disponibilidad de la información. Para ello, Mike Barrow, propietario de Costaflores, está valorando la posibilidad de aumentar la recompensa para aquellos validadores que se demuestre que almacenan una copia sincronizada de la *side chain* en sus dispositivos.

La recompensa extra podría consistir en hacerles entrega de una participación en la propiedad de la empresa Costaflores.

La conclusión es que una *side chain* puede aportar los valores buscados en una trazabilidad, como son la inmutabilidad de la información, el no repudio, la alta disponibilidad y la veracidad. Pero esta veracidad se dará si y solo si los sensores envían la información debidamente certificada y si los validadores realizan correctamente su trabajo. Para ello, es fundamental definir qué ganarán los validadores y dejar claro que la mala realización de su trabajo puede causar la pérdida de valor de las compensaciones recibidas, especialmente si han recibido acciones de la empresa.

Por lo tanto, se puede concluir que estamos ante una trazabilidad impregnada de VERDAD que fomentará la elaboración de un producto de consumo «extraordinario», que podrán disfrutar los consumidores en general y doblemente la comunidad OPENVINO. Esta comunidad está formada por el conjunto de trabajadores de Costaflores, por validadores de la *side chain*, por los propietarios de los tokens *winecoins* MTB* y por los posibles accionistas de la empresa Costaflores.



Figura 18. El centro de este TFM

Anexo 1. Presupuesto plataforma sensores



Vinduino R3

33,29 €



LM-210H

LoRaLAN, RID-LR01H para Vinduino con una frecuencia de 868MHz correspondiente a la banda Europea LoRa

18,99 €



Panel solar 5 V 200mA

8,99 €



Batería 3,7 V 900 mAh

6,83 €



Irrometer 200SS

Sensor Watermark para medir la humedad del suelo

150,72 €

37,68 euros unidad x 4 unidades



DS18B20

Sensor de temperatura resistente al agua

2,00 €

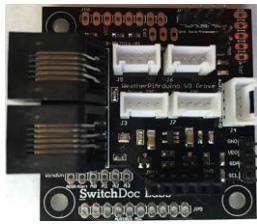


Adafruit FTDI Serial TTL-232 USB

Cable adaptador USB a serie

16,75 €

Anexo 2. Presupuesto estación meteorológica



WeatherPiArduino V3

Placa de interface entre los sensores de la estación meteorológica y el gateway Raspberry Pi.¹⁹

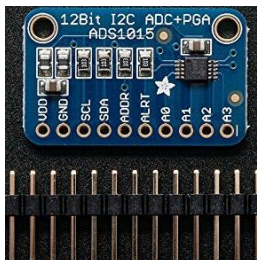
21,35 €



DS3231

Reloj *-clock-*, de bajo precio. Incorpora un oscilador de cristal y protocolo I2C. Asegura la precisión a largo plazo.²⁰

6,83 €



ADS1015

12-Bit ADC - 4 Amplificador de cuatro canales con amplificador de ganancia programable.²¹

10,21 €



AM2315

Sensor de temperatura y humedad con protocolo de comunicaciones I2C²²

21,31 €



SKU: 0015-WR-DSBT

Estructura para Anemómetro / Viento / Lluvia²³

64,88 €

¹⁹ <https://shop.switchdoc.com/collections/raspberry-pi/products/the-weather-board-w-grove-interface-board-for-weather-instruments-for-raspberry-pi-arduino>

²⁰ <https://shop.switchdoc.com/collections/raspberry-pi/products/switchdoc-labs-ds3231-at24c32-eprom-battery-and-software>

²¹ <https://adafruit.com/product/1083>

²² <https://shop.switchdoc.com/collections/raspberry-pi/products/grove-am2315-encased-i2c-temperature-humidity-sensor-for-raspberry-pi-arduino>

²³ <https://shop.switchdoc.com/collections/raspberry-pi/products/weatherrack-anemometer-wind-rain-for-weatherpiarduino-weatherplus-raspberry-pi-arduino>

Anexo 3. Presupuesto *gateway*



Raspberry Pi 3 Modelo B

34,45 €



Fuente de alimentación para Raspberry Pi

8,99 €



Tarjeta SanDisk Extreme 64GB

26,90 €

LD-20H



Artefacto LoRa USB para Vinduino con una frecuencia de 868MHz, para adaptarse a las especificaciones Europeas relacionadas con la banda de frecuencias LoRa.

80,56 €

Bibliografía

- Ambrosus, website. (2018). Decentralised IoT Networks for Next-Generation Supply Chains. Recuperado 2 de junio de 2018, de <https://ambrosus.com/technology/>
- Barrow, M. (2018a). MIKETANGOBRAVO18 (MTB18) ERC20 Token Tracker. Recuperado 2 de junio de 2018, de <https://etherscan.io/token/0x1bcfd19f541eb62c8cfebe53fe72bf2afc35a255>
- Barrow, M. (2018b). Publishing Vineyard Data - OpenVino - OpenVino. Recuperado 12 de junio de 2018, de <http://wiki.costaflores.com:8090/display/OP/Publishing+Vineyard+Data>
- Casey, M. J. (Trad.). (2018). In blockchain we trust. Recuperado 2 de junio de 2018, de <https://www.technologyreview.com/s/610781/in-blockchain-we-trust/>
- CIO, R. (2018, marzo 28). Tecnología IoT hace el control predictivo de los viñedos de la bodega Pago Aylés. Recuperado 2 de junio de 2018, de <http://www.ciospain.es/big-data/tecnologia-iot-hace-el-control-predictivo-de-los-vinedos-de-la-bodega-pago-ayles>
- Dannen, C. (2017). *Introducing Ethereum and Solidity: foundations of cryptocurrency and blockchain programming for beginners*. New York: Apress.
- Etapé, J. (2018). Blokchain storage proposal - OpenVino - OpenVino. Recuperado 4 de junio de 2018, de <http://wiki.costaflores.com:8090/display/OP/Blokchain+storage+proposal>
- Forroll, J. (2018). Aportaciones al TFM - OpenVino - OpenVino. Recuperado 3 de junio de 2018, de <http://wiki.costaflores.com:8090/display/OP/Aportaciones+al+TFM>
- González, T., & Gea, J. C. (2018). Solving the graphics of Daily Log HTML/CSS/JS - OpenVino - OpenVino. Recuperado 12 de junio de 2018, de <http://wiki.costaflores.com:8090/pages/viewpage.action?pageId=11993331>
- Hernández, N. (2016). «Tecno» viñedos: el Internet de las Cosas al servicio de la producción del vino | Life - ComputerHoy.com. Recuperado 2 de junio de 2018, de

<https://computerhoy.com/noticias/life/tecno-vinedos-internet-cosas-servicio-produccion-del-vino-54112>

Marr, B. (2018). How Blockchain Could End The Trade In Blood Diamonds - An Incredible Use Case Everyone Should Read. Recuperado 19 de junio de 2018, de <https://www.forbes.com/sites/bernardmarr/2018/03/14/how-blockchain-could-end-the-trade-in-blood-diamonds-an-incredible-use-case-everyone-should-read/>

Reinier van der Lee. (2016). Van der Lee Vineyard. Recuperado 2 de junio de 2018, de <http://vanderleevineyard.com/>

Rrisciotti, L. (2018). Use blockchain technology to tell the story of every product. Recuperado 2 de junio de 2018, de <https://www.dnvgl.com/mystory/mystory-project.html>

Schwab, K. (2016). *The fourth industrial revolution* (First U.S. edition). New York: Crown Business.

Tapscott, D., Tapscott, A., & Salmerón, J. M. (2017). *La revolución blockchain: descubre cómo esta nueva tecnología transformará la economía global*.

w3org. (2018). Paths – SVG 1.1 (Second Edition). Recuperado 12 de junio de 2018, de <https://www.w3.org/TR/SVG/paths.html#PathData>

WWF. (2018). Water Scarcity | Threats | WWF. Recuperado 2 de junio de 2018, de <https://www.worldwildlife.org/threats/water-scarcity>

Yaga, D. (2018). Draft NISTIR 8202, Blockchain Technology Overview, 59.