

**Mejora de Gestión de Servicios de TI a través de ITIL y la
Tecnología Blockchain**
**Improving IT Service Management using ITIL and Blockchain
Technology**



Trabajo Fin de Grado
Curso 2021-2022

José Andrés Velasco Santos
Yule Zhang

Dirigido Por:

María Cruz Valiente Blázquez

Grado en Ingeniería del Software
Facultad de Informática
Universidad Complutense de Madrid

**Mejora de Gestión de Servicios de TI a través de ITIL y la
Tecnología Blockchain**

**Improving IT Service Management using ITIL and Blockchain
Technology**

Trabajo de Fin de Grado en Ingeniería del Software
Departamento de Ing. Software e Inteligencia Artificial

José Andrés Velasco Santos
Yule Zhang

Dirigido Por:

María Cruz Valiente Blázquez

Convocatoria: Junio 2022

Grado en Ingeniería del Software
Facultad de Informática
Universidad Complutense de Madrid

30 de mayo de 2022

Agradecimientos

En primer lugar, queremos agradecer profundamente a nuestra directora del proyecto, Maricruz, por la propuesta que hizo para este proyecto, por las constantes ayudas que recibimos de su parte durante todo el curso, por mandarnos artículos de interés y contactos de diferentes personas, también por dedicar tiempo a reunirse con nosotros periódicamente para dar seguimiento al proyecto. Sus directrices e indicaciones nos han ayudado muchísimo para avanzar con el trabajo, sin ella es imposible llegar hasta dónde estamos ahora.

Además, queremos agradecer a nuestras familias, parejas y amigos, por su apoyo y ánimo constante.

Resumen

Mejora de Gestión de Servicios de TI a través de ITIL y la tecnología Blockchain

Siendo la clave para mejorar la calidad del servicio de TI en una empresa, ITIL es una guía de buenas prácticas para la gestión de servicios de TI y abarca toda la infraestructura, el desarrollo y las operaciones de TI.

Blockchain constituye una tecnología segura y transparente que utiliza la descentralización y el hashing criptográfico para aplicaciones en el mundo real. Puede ayudar a cada persona a tomar el control total de cada movimiento que realiza de la manera más segura posible.

En este proyecto se propone trabajar en una solución para mejorar la gestión de servicios de TI para empresas utilizando SLA, el aspecto más fundamental de ITIL, y la tecnología Blockchain.

Palabras clave: TI, Gestión de Servicios, ITIL, Blockchain, SLA

Abstract

Improving IT Service Management using ITIL and Blockchain technology

Being the key to improve the quality of IT service in a company, ITIL is a guide of good practices for IT service management and covers all IT infrastructure, development and operations.

The Blockchain is the most secure and transparent technology by using decentralization and cryptographic hashing in today's world. It can help every person take total control of every single move that he makes in the safest way possible.

In this project we propose to work for a solution to improve IT service management for companies using SLA, the most fundamental aspect of ITIL, and Blockchain technology.

Keywords: IT, Service Management, ITIL, Blockchain, SLA

Índice de contenidos

Capítulo 1 - Introducción	16
1.1. Motivación	16
1.2. Objetivos	17
1.3. Plan de trabajo	18
Figura 1.1 Cronograma de trabajo	18
1.4. Estructura de la memoria	19
Chapter 1 - Introduction	20
1.1. Motivation	20
1.2. Objectives	21
1.3. Workplan	22
Figure 1.1 Work schedule	22
1.4. Memory structure	23
Capítulo 2 - Estado de la cuestión	24
2.1. Introducción al ITIL	24
2.1.1. ¿Qué es ITIL?	24
2.1.2. ¿Cómo funciona ITIL?	24
Figura 2.1 Ciclo de vida del Servicio	25
2.1.3. Diseño de servicio de ITIL	25
2.1.4. ¿Qué es la Gestión del nivel de servicio?	26
2.1.5. Introducción a los SLAs	26
Figura 2.2 Relación SLA y ITIL	26
Tabla 2.1 Información contenida en un SLA	28
2.1.6. Los KPIs dentro de ITIL	28
2.2. Tecnología Blockchain	29
2.2.1. Introducción a la Blockchain	29
Figura 2.3 Cómo actúa la Blockchain	30
Figura 2.4 Estructura de aplicaciones de Web2	31
Figura 2.5 Estructura de aplicaciones de Web2	32
2.2.2. Conceptos importantes dentro de la Blockchain	33
2.2.3. Ventajas de la Blockchain	35
2.2.4. Desventajas de la Blockchain	36
Capítulo 3 - Tecnologías utilizadas	37
3.1. Javascript	37
3.2. React	37
3.3. GitHub	37

3.4. WebStorm	38
Figura 3.1 WebStorm	38
3.5. Postman	38
Figura 3.2 Postman	39
3.6. Ethereum	39
3.6.1. ¿Qué es Ethereum?	39
3.6.2. ¿Cómo funciona Ethereum?	40
3.6.3. Transacciones en Ethereum	40
3.6.4. Estructura de Ethereum	41
3.7. Remix	41
Figura 3.3 Remix	42
3.8. Solidity	42
3.9. SQL	42
3.10. MySQL	42
3.11. Express	43
3.12. GraphQL	43
3.13. The Graph	43
3.13.1. ¿Qué es The Graph?	43
3.13.2. ¿Cómo funciona The Graph?	44
3.13.3. ¿Qué es un subgrafo?	44
3.13.4. ¿Qué funciones existen dentro de la red?	44
Capítulo 4 - Sistema descentralizado basado en Blockchain que almacena SLAs, SLink	45
4.1. Nombre	45
4.2. Arquitectura	45
Figura 4.1 Diagrama de arquitectura del sistema	47
4.3. Front-end	48
4.3.1. Página principal	48
Figura 4.2 Página principal	49
4.3.2. Página de SLA y elementos de SLA en SLink	49
Tabla 4.1 Elementos de SLA en SLink	50
Figura 4.3 Página de SLA (visitante)	51
Figura 4.4 Información de error (visitante)	52
Figura 4.5 Página de SLA (cliente) - parte 1	54
Figura 4.6 Página de SLA (cliente) - parte 2	55
4.3.3. Conectar con el monedero digital	56
Figura 4.7 Botón Conectar monedero	56
Figura 4.8 Elegir monedero	56
Figura 4.9 Notificación MetaMask	57

Figura 4.10 Selección de cuenta Ethereum en MetaMask	58
4.3.4. Creación de un SLA	58
Figura 4.11 Confirmación de pago de tasas de transacción Ethereum - paso 1	60
Figura 4.12 Confirmación de pago de tasas de transacción Ethereum - paso 2	61
Figura 4.13 Confirmación de pago de tasas de transacción Ethereum - paso 3	62
4.3.5. Lista de entidades	63
Figura 4.14 Listado de clientes	64
Figura 4.15 Creación de cliente	65
Figura 4.16 Actualización de cliente	66
Figura 4.17 Listado de empresas	67
Figura 4.18 Creación de empresa	68
Figura 4.19 Actualización de empresa	69
Figura 4.20 Listado de SLAs	70
Figura 4.21 Listado de datos de un SLA	71
Figura 4.22 Listado de peticiones de contacto	72
4.3.6. Dashboard	73
Figura 4.23 Sección general - parte 1	75
Figura 4.24 Sección general - parte 2	76
Figura 4.25 Sección servicios (un ejemplo)	77
4.3.7. Idiomas	78
Figura 4.26 Cambio de idioma	79
Figura 4.27 Cambio de idioma (resultado)	80
4.3.8. Página de Aviso legal y política de privacidad	81
Figura 4.28 Aviso legal y política de privacidad	82
Figura 4.29 Acceder a Aviso legal y política de privacidad (forma 1)	82
Figura 4.30 Acceder a Aviso legal y política de privacidad (forma 2)	83
4.3.9. Página de Conocernos	83
Figura 4.31 Página Conocernos	84
Figura 4.32 Formulario de contacto	85
4.4. Contrato inteligente	85
4.4.1. Estructuras	86
Figura 4.33 Diagrama del modelo de datos del contrato inteligente	86
4.4.2. Variables	86
Tabla 4.2 Variable “Proveedor”	87
Tabla 4.3 Variable “Niveles de servicio”	87
Tabla 4.4 Variable “IDs de servicios”	87
Tabla 4.5 Variable “Servicios”	87
Tabla 4.6 Variable “IDs de servicios extra”	88
Tabla 4.7 Variable “Servicios extra”	88

Tabla 4.8 Variable “IDs de espacios de servicios”	88
Tabla 4.9 Variable “Espacios de servicios”	89
Tabla 4.10 Variable “IDs de licencias”	89
Tabla 4.11 Variable “Licencias”	89
Tabla 4.12 Variable “IDs de periodicidades de informes de revisión”	89
Tabla 4.13 Variable “Periodicidades de informes de revisión”	90
Tabla 4.14 Variable “IDs de periodicidades de facturación”	90
Tabla 4.15 Variable “Periodicidades de facturación”	90
Tabla 4.16 Variable “IDs de métodos de facturación”	91
Tabla 4.17 Variable “Métodos de facturación”	91
Tabla 4.18 Variable “IDs de SLAs”	91
Tabla 4.19 Variable “SLAs”	91
4.4.3. Funciones	92
Tabla 4.20 Funciones del contrato inteligente	93
4.4.4. Eventos	93
Tabla 4.21 Eventos del contrato inteligente	94
4.5. Back-end	94
4.5.1. Base de datos privada	95
Figura 4.34 Diagrama relacional de la base de datos	97
4.5.2. The Graph en el back-end	97
Figura 4.35 Diagrama de clases de The Graph	98
4.5.3. API REST	98
Tabla 4.22 API REST para gestión de clientes	100
Tabla 4.23 API REST para gestión de empresas	100
Tabla 4.24 API REST para gestión de SLAs	101
Tabla 4.25 API REST para gestión de peticiones de contacto	102
Tabla 4.26 API REST para gestión de servicios de SLA	102
Tabla 4.27 API REST para gestión de servicios extra de SLA	103
Tabla 4.28 API REST para gestión de espacios de servicios de SLA	103
Tabla 4.29 API REST para gestión de licencias de SLA	104
Tabla 4.30 API REST para gestión de reportes de revisión de SLA	104
Tabla 4.31 API REST para gestión de facturación de SLA	104
Tabla 4.32 API REST para gestión de métodos de facturación de SLA	105
4.6. Despliegue del sistema	105
4.6.1. Despliegue del contrato inteligente	105
Figura 4.36 Compilar el contrato inteligente	106
Figura 4.37 MetaMask con Rinkeby	107
Figura 4.38 Opciones de despliegue	107
Figura 4.39 Funciones y variables del contrato inteligente desplegado	108

Figura 4.40 Llamada a una función del contrato inteligente desplegado	109
4.6.2. Despliegue del subgrafo de The Graph	109
Figura 4.41 Botón de creación de un subgrafo	110
Figura 4.42 Opciones a configurar en la creación de un subgrafo	110
Figura 4.43 Datos del subgrafo	111
Figura 4.44 Verificación y publicación del código de un contrato inteligente	112
Figura 4.45 Código verificado y publicado de un contrato inteligente	113
Figura 4.46 Instalación de The Graph CLI	114
Figura 4.47 Inicialización del subgrafo	114
Figura 4.48 Generación y construcción del código de un subgrafo	115
Figura 4.49 Autenticación y despliegue de un subgrafo	115
Figura 4.50 Página de edición de un subgrafo donde se muestran varios datos de este	115
Figura 4.51 Configuración para la publicación de un subgrafo	116
Figura 4.52 Subgrafo publicado a la espera de ser indexado	117
4.6.3. Despliegue del back-end en local	117
4.6.4. Despliegue del front-end en local	118
4.7. Licencia y código fuente	119
4.7.1. Licencia	119
Figura 4.53 Licencia	119
4.7.2. Código fuente	119
Capítulo 5 - Trabajo individual	120
5.1. Trabajo individual de José Andrés	120
5.2. Trabajo individual de Yule	122
Capítulo 6 - Conclusiones y trabajo futuro	124
6.1. Conclusiones	124
6.2. Trabajo futuro	127
Chapter 6 - Conclusions and future work	129
6.1. Conclusions	129
6.2. Future work	131
Bibliografía	134

Capítulo 1 - Introducción

1.1. Motivación

Hoy en día, la necesidad del departamento de la Tecnología de la Información (TI) sigue creciendo, y lo está mostrando de una forma innovadora. Como afirma el artículo *¿Hace falta un departamento de informática? Pues, depende*[3], en la actualidad, se está viendo el impacto del departamento de TI en el resto de los departamentos.

“Algunas de estas cosas ya están pasando y, en muchas organizaciones los que más saben de Salesforce están en los departamentos de ventas, los que más saben de SAP están en finanzas o los que saben de las herramientas de datos están en todas partes en modo de autoservicio. Algunos departamentos de ‘innovación’ o de investigación y desarrollo están llenos de informáticos, afortunadamente.”

Por lo que se puede ver que la TI ya no solo constituye una parte importante en una empresa, sino que cada vez se está haciendo más imprescindible.

Además, a raíz de la pandemia de la COVID-19 y la nueva ola de teletrabajo, la transformación digital en las empresas en el área de TI se ha acelerado de manera muy significativa, produciéndose cada vez mayor demanda de servicios.

Por otro lado, debido a la crisis sanitaria y económica que se está viviendo, se están cerrando muchas empresas en el mercado. Se puede ver en la siguiente noticia *El tejido productivo no se recupera: España tiene hoy 77.831 empresas menos que antes de la pandemia*[4] del periódico *El Mundo*:

“No todas las empresas que tuvieron que bajar la persiana en España por la crisis de la pandemia, han podido volver a subirla. De hecho, a cierre de 2021 el país cuenta con 1.411.902 empresas dadas de alta en la Seguridad Social, frente a las 1.489.733 que tenía en febrero de 2020 antes de que irrumpiera con fuerza la pandemia.”

En este sentido, los entornos empresariales son cada vez más competitivos y las expectativas de la alta dirección son cada vez más ambiciosas en lo que respecta a los servicios de TI, tanto a nivel interno como externo (es decir, servicios que desea ofrecer a sus clientes).

La Biblioteca de Infraestructura de Tecnologías de Información (del inglés *Information Technology Infrastructure Library*, ITIL) puede ayudar a las empresas a cumplir sus objetivos

ofreciendo unos servicios de TI con mayor calidad. ITIL ha sido reconocida internacionalmente como guía de buenas prácticas para la gestión de servicios de TI y es utilizada por cientos de organizaciones a nivel mundial demostrando su eficacia.

La adopción de los procesos de ITIL se hace necesaria para poder obtener servicios TI con calidad y una alineación con el negocio en las empresas, ya que los directivos de las empresas esperan que las necesidades de la organización se conviertan en soluciones, y que TI no constituya un freno al negocio, como suele ocurrir en ciertas ocasiones.

Por otro lado, los servicios descentralizados basados en la tecnología Blockchain, aunque todavía no son lo suficientemente maduros, están demostrando su gran potencial en la Industria gracias a la seguridad en su infraestructura, su transparencia y su trazabilidad.

Es por ello, que en este proyecto se pretende hacer uso de ITIL en combinación con la tecnología Blockchain, a través de la implementación de contratos inteligentes (del inglés *smart contracts*), para que distintas organizaciones, independientemente de su tamaño y condición, puedan gestionar los Acuerdos de Nivel de Servicio (del inglés *Service Level Agreements*, SLAs) con sus clientes de una manera transparente, segura y descentralizada asegurando la calidad de la gestión del servicio.

Cabe destacar que en el momento de la escritura de este proyecto no se ha encontrado ningún trabajo que vincule ITIL con la tecnología Blockchain para obtener los beneficios que pueden ofrecer la fusión de estos dos dominios.

1.2. Objetivos

El objetivo principal de este proyecto es la creación de un sistema descentralizado basado en la tecnología Blockchain (conocido como Web3¹), que favorezca la gestión de los SLAs que se establecen entre los proveedores de servicios de TI y sus clientes siguiendo las buenas prácticas de ITIL.

A partir de este objetivo principal, se establecen los siguientes objetivos específicos:

- Implementar el sistema a través de una interfaz que sea lo más intuitiva, cómoda y accesible posible. Para que, de esta manera, el sistema pueda llegar al máximo público.

¹ <https://www.gartner.es/es/articulos/que-es-la-web3>

- Establecer y mejorar la relación y comunicación entre proveedores de servicios de TI y los clientes de manera segura y transparente a partir de los SLAs.
- Garantizar que tanto los proveedores de servicios de TI como los clientes tengan una expectativa clara y sin ambigüedades del nivel de servicio que se va a contratar.
- Facilitar la contratación y monitorización de los servicios de TI por parte de los clientes a través de las transacciones registradas en Blockchain, mediante la implementación de contratos inteligentes gestionados con monederos digitales.
- Favorecer la medición y revisión de los niveles de servicio contratados mediante la implementación de un cuadro de mando (del inglés *dashboard*) que recupere los datos de contratos inteligente registrados en Blockchain, y muestre los principales Indicadores Clave de Rendimiento (del inglés *Key Performance Indicators*, KPIs) vinculados a los SLAs.
- Adquirir nuevos conocimientos en estándares y tecnologías punteras que permitan afrontar con mayor garantía de éxito la vida profesional como graduados universitarios informáticos.

1.3. Plan de trabajo

En la [Figura 1.1](#) se describen las fases del proyecto y el tiempo dedicado para cada una de ellas:



Figura 1.1 Cronograma de trabajo

Se resumen las respectivas fases del plan de trabajo a continuación:

1. **Primera Toma de Contacto:** Esta ha sido la primera fase realizada en este proyecto, que ha consistido en tener una visión inicial sobre el proyecto, definir los objetivos del mismo y conocer las tecnologías que se van a utilizar.

2. **Investigaciones:** Durante esta fase, se han realizado investigaciones sobre las diferentes tecnologías, términos y posibles implementaciones de funcionalidades del sistema objetivo. El resultado de esta fase está reflejado en el [Capítulo 2 Estado de la cuestión](#) de este documento.
3. **Desarrollo Sistema:** Esta fase se ha enfocado en el desarrollo del sistema, tanto en lo que respecta a la parte del front-end como del back-end. Las tecnologías utilizadas para su implementación se mencionan en el [Capítulo 3 Tecnologías utilizadas](#).
4. **Dashboard:** En esta fase se ha desarrollado un dashboard intuitivo que permite visualizar la información clave relativa a los SLAs mediante la aplicación de ciertos KPIs.
5. **Pasos Finales:** Los pasos finales del proyecto se han centrado principalmente en completar la memoria, así como en mejorar pequeños aspectos del sistema implementado.

1.4. Estructura de la memoria

El resto de la memoria está estructurada de la siguiente manera:

[Capítulo 2 Estado de la cuestión:](#) Este capítulo contempla todos los conocimientos importantes que se requieren para la realización del sistema. Para ello, por un lado, se explica ITIL y su aspecto más fundamental de cara al ámbito de este proyecto, los SLAs. Por otro lado, en este capítulo se introduce la tecnología Blockchain, así como sus términos principales.

[Capítulo 3 Tecnologías utilizadas:](#) En este capítulo se presentan todas las tecnologías que se han utilizado para este Trabajo Fin de Grado (TFG).

[Capítulo 4 Sistema descentralizado basado en Blockchain que almacena SLAs, SLink:](#) En este capítulo se presenta en detalle el sistema desarrollado en este trabajo.

[Capítulo 5 Trabajo individual:](#) En este capítulo se detalla el trabajo realizado por cada uno de los autores del presente proyecto.

[Capítulo 6 Conclusiones y trabajo futuro:](#) En este capítulo se describen las conclusiones extraídas del proyecto, así como las futuras líneas de investigación si se continuará el desarrollo.

Chapter 1 - Introduction

1.1. Motivation

Nowadays, the need for the Information Technology (IT) department continues increasing, and it is showing in an innovative way. As stated in the article *Is an IT department necessary? Well, it depends [3]*, at present, the impact of the IT department is being seen in the rest of the departments.

“Some of these things are already happening and, in many organizations those who know the most about Salesforce are in the sales departments, those who know the most about SAP are in finance, or those who know the data tools are everywhere in self-service mode. Fortunately, some ‘innovation’ or research and development departments are full of computer scientists.”

It can be seen that IT is not only an important part of a company, but is becoming more and more essential.

In addition, as a result of the COVID-19 pandemic and the new wave of teleworking, the digital transformation in companies in the IT area has accelerated very significantly, producing an increasing demand for services.

On the other hand, due to the current health and economic crisis, many companies in the market are closing. It can be seen in the following news *The productive fabric is not recovering: Spain today has 77,831 fewer companies than before the pandemic[4]* from the newspaper *El Mundo*:

“Not all the companies that had to lower the blind in Spain due to the pandemic crisis, have been able to raise it again. In fact, at the end of 2021 the country had 1,411,902 companies registered with Social Security, compared to the 1,489,733 it had in February 2020 before the pandemic broke out with force.”

In this sense, business environments are increasingly competitive and the expectations of senior management are increasingly ambitious regarding IT services, both internally and externally (that is, services that they want to offer to their customers).

The Information Technology Infrastructure Library (ITIL) can help companies meet their goals by providing IT services with higher quality. ITIL has been internationally recognized as a best

practice guide for IT service management and is used by hundreds of organizations worldwide, proving its effectiveness.

The adoption of ITIL processes is necessary in order to obtain IT services with quality and alignment with the business in companies, since company managers expect that the needs of the organization become solutions, and that IT does not constitute a brake on business, as often happens on certain occasions.

On the other hand, decentralized services based on Blockchain technology, although they are not yet mature enough, are showing their great potential in the Industry thanks to the security of their infrastructure, transparency and traceability.

For this reason, this project intends to make use of ITIL in combination with Blockchain technology, through the implementation of smart contracts, so that different organizations, regardless of their size and condition, can manage Service Level Agreements (SLAs) with their clients in a transparent, secure and decentralized way, ensuring the quality of service management.

It should be noted that at the time of writing this project, no work has been found linking ITIL with Blockchain technology to obtain the benefits from the merger of these two domains.

1.2. Objectives

The main objective of this project is the creation of a decentralized system based on Blockchain technology (known as Web3²), which favors the management of SLAs established between IT service providers and their clients following good practices of ITIL.

As of this main objective, the following specific objectives are established:

- Implement the system through an interface that is as intuitive, comfortable and accessible as possible. So that, in this way, the system can reach the maximum public.
- Establish and improve the relationship and communication between IT service providers and customers in a secure and transparent way by means of SLAs.
- Ensure that both IT service providers and customers have a clear and unambiguous expectation of the service level to be contracted.

² <https://hbr.org/2022/05/what-is-web3>

- Facilitate the hiring and monitoring of IT services by customers via transactions recorded in Blockchain, by means of the implementation of smart contracts managed with digital wallets.
- Promote the measurement and review of contracted service levels by implementing a dashboard that retrieves data from smart contracts registered in Blockchain, and displays the main Key Performance Indicators (KPIs) linked to SLAs.
- Acquire new knowledge in standards and cutting-edge technologies that allow to face the professional life for computer science graduates with a greater guarantee of success.

1.3. Workplan

The [Figure 1.1](#) describes the phases of the project and the time dedicated to each of them:



Figure 1.1 Work schedule

The respective phases of the work plan are summarized below:

1. **First Contact:** This has been the first phase carried out in this project, which consists of having an initial vision of the project, defining its objectives and learning the technologies that are going to be used.
2. **Investigations:** During this phase, investigations on the different technologies, terms and possible implementations of functionalities of the target system have been carried out. The result of this phase is reflected in [Capítulo 2 Estado de la cuestión](#) of this document.
3. **System Development:** This phase has focused on the development of the system, both in terms of the front-end and the back-end. The technologies used for its implementation are mentioned in [Capítulo 3 Tecnologías utilizadas](#).

4. **Dashboard:** In this phase, an intuitive dashboard that allows visualizing the key information related to the SLAs through the application of certain KPIs has been developed.
5. **Final Steps:** The final steps of the project have focused mainly on completing the memory, as well as improving small aspects of the implemented system.

1.4. Memory structure

The rest of the memory is structured as the follow way:

Capítulo 2 Estado de la cuestión: This chapter covers all the important knowledge required to implement the system. To do this, on the one hand, it explains ITIL and its most fundamental aspect for the scope of this project, which are the SLAs. On the other hand, it introduces Blockchain technology, as well as its main terms.

Capítulo 3 Tecnologías utilizadas: This chapter presents all the technologies that have been used for this Final Degree Project (FDP).

Capítulo 4 Sistema descentralizado basado en Blockchain que almacena SLAs, SLink: This chapter presents in detail the system developed in this project.

Capítulo 5 Trabajo individual: This chapter details the work carried out by each of the authors of this project.

Chapter 6 Conclusions and future work: This chapter describes the conclusions drawn from the project, as well as future lines of research if its development continues.

Capítulo 2 - Estado de la cuestión

Como ya se ha comentado anteriormente en el presente documento, el objetivo principal de este proyecto es la creación de un sistema descentralizado basado en la tecnología Blockchain que facilite la gestión de SLAs entre clientes y proveedores de servicios de TI.

Por tanto, en este capítulo, por un lado, para la parte de gestión de servicios de TI, se realiza una introducción de los conceptos de ITIL; y por otro lado, para la parte de descentralización, se detalla la tecnología Blockchain.

2.1. Introducción al ITIL

2.1.1. ¿Qué es ITIL?

Como ya se ha explicado anteriormente en la sección [1.1. Motivación](#), ITIL se corresponde con las siglas en inglés de *Information Technology Infrastructure Library*.

ITIL abarca toda la infraestructura, desarrollo y operaciones de TI, y constituye el estándar de facto para la gestión de servicio de TI en las organizaciones.

2.1.2. ¿Cómo funciona ITIL?

El ciclo de vida del servicio en ITIL consta de las siguientes fases según el libro *Fundamentos de la Gestión de Servicios de TI basada en ITIL[2]*:

1. **Estrategia de servicio:** Facilita a las organizaciones a establecer metas y estrategias para cumplir los requisitos y prioridades de los clientes.
2. **Diseño del servicio:** Ayuda al diseño de los procesos y funciones e incluye el diseño de la tecnología, la infraestructura y los productos de la gestión del servicio de TI.
3. **Transición del servicio:** Se pretende encontrar un nuevo cambio organizacional a la vez que se mantienen los servicios actuales para tener una opción B de cara a los riesgos.
4. **Operación del servicio:** Garantiza que las tareas operacionales diarias no se interrumpan con el fin de generar valor para los clientes y los proveedores de los servicios.
5. **Mejora continua de los servicios:** Se centra en la mejora continua de los procesos.

Las empresas pueden adoptar algunas de estas etapas, siempre y cuando sean adecuadas para la organización. Esto hace que ITIL sea flexible y pueda adaptarse según las distintas circunstancias. La [Figura 2.1](#) muestra la visión del ciclo de vida del Servicio de acuerdo a ITIL.



Figura 2.1 Ciclo de vida del Servicio³

2.1.3. Diseño de servicio de ITIL

Dentro de la etapa de diseño de servicio de ITIL se obtienen múltiples procesos (según el libro *Fundamentos de la Gestión de Servicios de TI basada en ITIL[2]*):

- Gestión del catálogo de servicio
- Gestión del nivel de servicio
- Gestión de la capacidad
- Gestión de la disponibilidad
- Gestión de la continuidad ITSCM (Gestión de Continuidad del Servicio de TI)
- Gestión de la seguridad de la información
- Gestión de suministradores

³ <https://consultora-impresajimdofree.com/m%C3%A9todos/frameworks/itil/>

Entre todas las etapas disponibles de ITIL, nos centraremos en la etapa de diseño de servicio, y sobre todo su proceso de **Gestión del nivel de servicio**, donde se habla de los SLAs.

2.1.4. ¿Qué es la Gestión del nivel de servicio?

Como bien se define en el libro *Fundamentos de la Gestión de Servicios de TI basada en ITIL[2]*, la gestión del nivel de servicio tiene como objetivo garantizar que el servicio de TI proporcionado por la empresa tiene un nivel adecuado, mantener y mejorar la calidad de los servicios TI a un precio aceptable.

Dentro de la gestión del nivel de servicio se puede encontrar el concepto SLA (Acuerdos de nivel de servicio, del inglés *Service Level Agreement*). Se centrará en el cual en este proyecto, que se trata de un acuerdo escrito entre el proveedor de servicio de TI y el cliente. En las siguientes secciones se darán más detalles sobre ello.

2.1.5. Introducción a los SLAs

Como explicado en la sección anterior, un SLA es un conjunto de promesas hechas por el proveedor de servicios de TI al cliente, definiendo las metas deseadas del servicio de TI y las responsabilidades de ambas partes.

Se puede observar la [Figura 2.2](#) para entender mejor la relación entre los SLAs e ITIL.

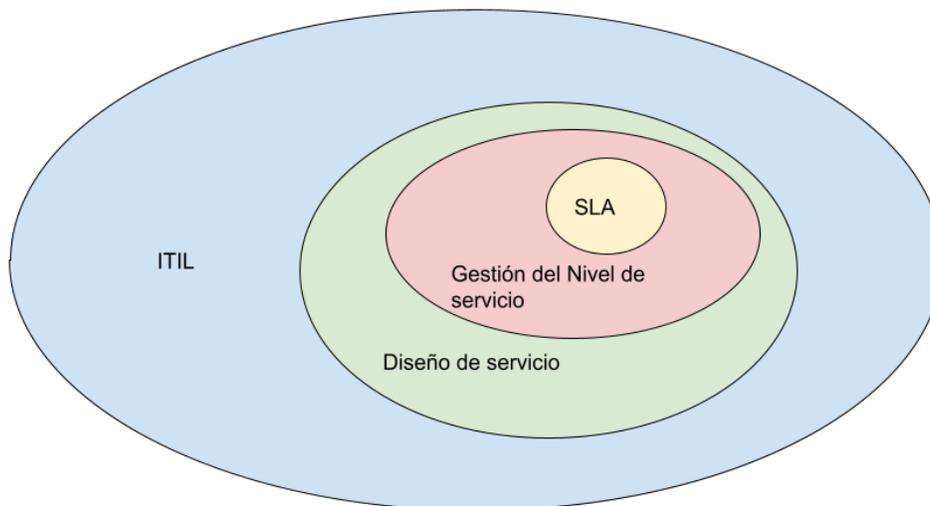


Figura 2.2 Relación SLA y ITIL

Los elementos que pueden formar parte de un SLA son los que se muestran en la [Tabla 2.1](#) (ver el capítulo 6.4.9.1 del libro *Guía Práctica de ISO/IEC 20000-1 para Servicios TIC (2A. ED.)*[1]).

Atributos	Contenido
Partes contratantes	Datos de los participantes en el acuerdo: cliente (DNI, nombre y apellidos, nacionalidad, sexo, ciudad, provincia, nombre de empresa, dirección de empresa, código fiscal de empresa), proveedor (DNI, nombre y apellidos, nacionalidad, sexo, ciudad, provincia, nombre de empresa, dirección de empresa, código fiscal de empresa).
Definiciones	Proveedor: parte contratante que ofrece servicios Cliente: parte contratante que contrata servicios.
Duración del SLA	Periodo de vigencia del acuerdo según las partes contratantes (fecha de inicio y fin).
Servicios cubiertos	Servicios prestados por el proveedor al cliente.
Comunicación	Números de teléfonos y/o email de personas de contacto para este SLA junto con su nombre.
Documentos relacionados	Contratos y cualquier otro documento necesario para prestar y gestionar el servicio.
Horario del servicio	Horarios en los que se prestará el servicio (hora de inicio y fin).
Niveles de servicio	Parámetros bajo los que se prestará el servicio.
Procedimiento de reclamación	Medios a disposición del cliente para presentar reclamaciones.
Exclusiones	Casos no considerados para el cálculo de los parámetros de los niveles de servicio.
Incentivos/penalización del rendimiento	Sistema de penalizaciones o incentivos que se sigue en función del cumplimiento mejor o peor con los niveles de servicio.
Cambio	Canales de comunicación y gestión de cambios entre proveedor y cliente.
Continuidad y seguridad de los servicios	Medidas de seguridad aplicadas al servicio para garantizar los parámetros de seguridad necesarios.

Atributos	Contenido
Revisión e informes de los servicios	Informes a emitir: periodicidad, contenido y destinatarios.
Gestión de incidencias en la prestación del servicio	Descripción de la gestión de incidencias, incluyendo aplicaciones, acciones de detección, registro y escalado en la prestación del servicio.
Confidencialidad	Cláusulas de confidencialidad necesarias.
Monitorización	Periodicidad y métodos mediante los que se comprobará el nivel de cumplimiento.
Facturación	Periodos y métodos de facturación.

Tabla 2.1 Información contenida en un SLA

De todos estos elementos, en este proyecto se han considerado los más relevantes: (se dará más detalle en la sección [4.3.2. Página de SLA y elementos de SLA en SLink](#)).

- Partes contratantes
- Duración del SLA
- Servicios cubiertos
- Horario de servicio
- Licencias
- Niveles de servicio
- Facturación

2.1.6. Los KPIs dentro de ITIL

Los Indicadores Clave de Rendimiento (del inglés *Key Performance Indicators*, KPIs) se suelen utilizar principalmente para medir el nivel de servicio, es decir, evaluar si los procesos de ITIL funcionan como las expectativas. Como indican en el capítulo 10 del libro *Measuring ITIL: Measuring, Reporting and Modeling - The IT Service Management Metrics That Matter Most to IT Senior Executives*[12], “Los KPIs son fundamentales para gestionar y controlar las actividades de gestión del nivel de servicio”. También se afirma en *Revisión y propuesta de indicadores (KPI) de la Biblioteca en los medios sociales*[5]:

“KPIs son frecuentemente utilizados para ‘valorar’ actividades complicadas de medir, como los beneficios de desarrollos líderes, compromiso de empleados, servicio o satisfacción.”

En la sección 10.2 del libro *Measuring ITIL: Measuring, Reporting and Modeling - The IT Service Management Metrics That Matter Most to IT Senior Executives*[12], se ha mencionado 9 KPIs recomendados:

1. Calificación general de satisfacción del cliente.
2. Tasa de cobertura de SLA.
3. Tasa de cobertura OLA (Acuerdo de Nivel Operativo, del inglés, *Operational Level Agreement*).
4. Porcentaje de servicios de proveedores prestados sin objetivos de servicio acordados.
5. Total de penalizaciones por servicio pagadas.
6. Porcentaje de objetivos de servicio de SLA cumplidos.
7. Porcentaje de SLA con propietarios de servicios responsables.
8. Nivel de soporte de herramientas de administración de nivel de servicio.
9. Madurez del proceso de gestión del nivel de servicio.

En este proyecto, debido a la limitación del tiempo y alcance, y la dificultad de probar el proyecto en los entornos empresariales reales, se han de definir unos KPIs propios y prácticos. Sobre dichos KPIs se dará más detalle en la sección [4.3.6. Dashboard](#). Hay que tener en cuenta que habría que definir KPIs como los mencionados anteriormente en entornos reales (ver la sección [6.2. Trabajo futuro](#)).

2.2. Tecnología Blockchain

2.2.1. Introducción a la Blockchain

IBM⁴ define la Blockchain (en castellano, cadena de bloques) como:

“Blockchain es un libro mayor compartido e inmutable que facilita el proceso de registro de transacciones y de seguimiento de activos en una red de negocios. Un activo puede ser tangible (una casa, un auto, dinero en efectivo, terrenos) o intangible (propiedad intelectual, patentes, derechos de autor, marcas). Prácticamente cualquier cosa de valor

⁴ <https://www.ibm.com/es-es>

puede ser rastreada y comercializada en una red de blockchain, reduciendo el riesgo y los costos para todos los involucrados.”

La Blockchain (conocido como Web3⁵) es “una forma de crear un libro mayor distribuido, robusto, seguro y transparente”, como se indica en el libro *Economics of Blockchain*[8]. No necesita la intermediación de terceros. En la [Figura 2.3](#) se muestra un ejemplo sencillo.

Tradicionalmente cuando una persona A quiere hacer una transferencia a una persona B, los bancos de las personas A y B son los encargados de la gestión. Ni la persona A ni B tiene control sobre el proceso de la transferencia, del que solo los bancos tienen toda la información. Ambas personas dependen de sus bancos y de su forma de hacer las cosas para completar la transacción. Están sujetas a sus condiciones y comisiones.

Como indican en el *Bitcoin: A Peer-to-Peer Electronic Cash System*[10], lo cual trae como consecuencia:

1. Problema de confianza de los clientes con la entidad bancaria.
2. Alto coste de los intermediarios.
3. Un cierto porcentaje de fraude.

Pero con la Blockchain, la transferencia llega desde la persona A directamente a la persona B, eliminando los intermediarios y descentralizando toda la gestión. De este modo, se asegura que las transacciones sean seguras, confiables e irreversibles y se ahorra el coste de los intermediarios.

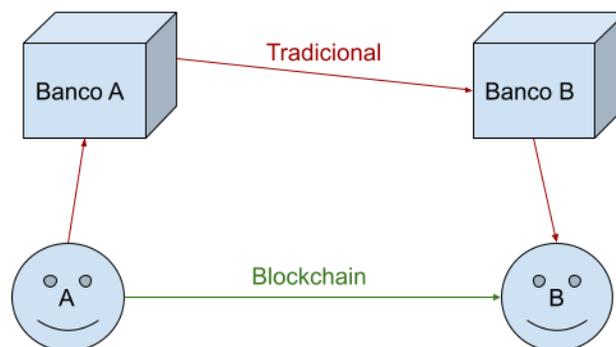


Figura 2.3 Cómo actúa la Blockchain

⁵ <https://www.gartner.es/es/articulos/que-es-la-web3>

¿Qué diferencia hay entre las aplicaciones de Web3 (o “DApps”, del inglés *Decentralized Application*) y de Web2? En primer lugar, hay que entender qué son las aplicaciones de Web2, básicamente son las aplicaciones web que permiten a los usuarios interactuar y colaborar en línea. En la [Figura 2.4](#) se muestra un ejemplo de estructura típica de aplicaciones de Web2.

Se puede observar que las aplicaciones de Web2 tienen las siguientes capas:

1. **Base de datos:** Una base de datos constantemente actualizada para almacenar datos esenciales.
2. **Back-end:** Código del back-end para describir la lógica de la aplicación.
3. **Front-end:** Código del front-end para describir la interfaz de la aplicación.

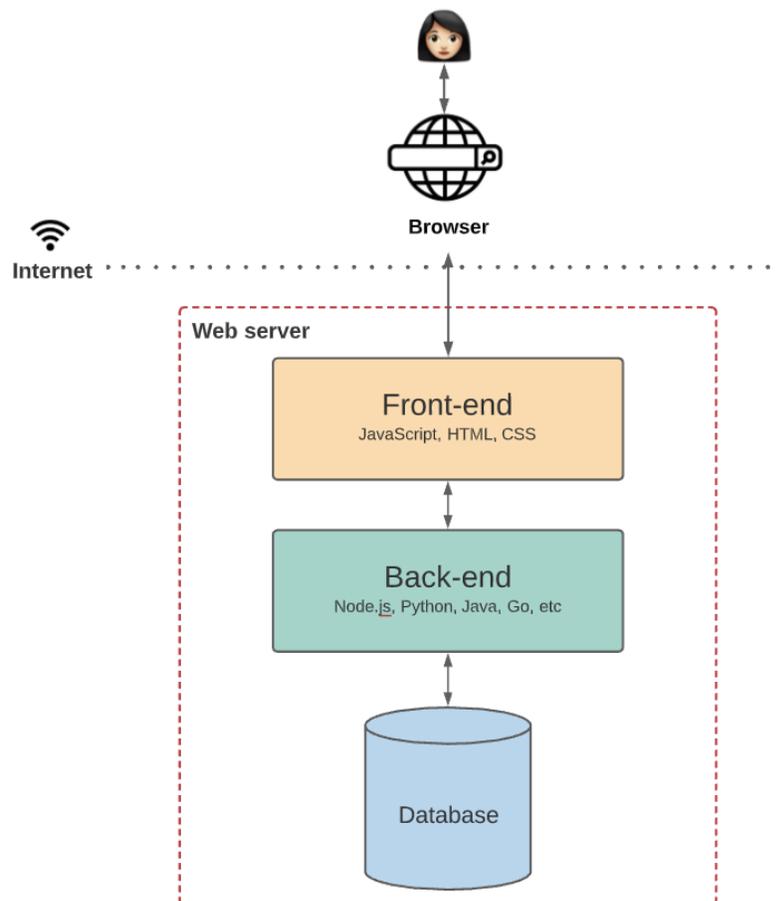


Figura 2.4 Estructura de aplicaciones de Web2⁶

⁶ <https://www.preethikasireddy.com/post/the-architecture-of-a-web-3-0-application>

Sin embargo, las aplicaciones de Web3 son totalmente distintas:

1. No tienen una base de datos centralizada para almacenar datos de la aplicación, y tampoco disponen de un servidor web donde se incluye la lógica del back-end.
2. Hay proveedores de servicio que permiten la conexión entre el cliente (front-end) a la Blockchain (MetaMask, por ejemplo) pero se necesita la firma del cliente utilizando su propia clave privada.
3. Se permite recuperar datos de la Blockchain utilizando The Graph (ver la sección [3.13. The Graph](#) para conocer en detalle The Graph).

Finalmente, la estructura de aplicaciones de Web3 es la siguiente:

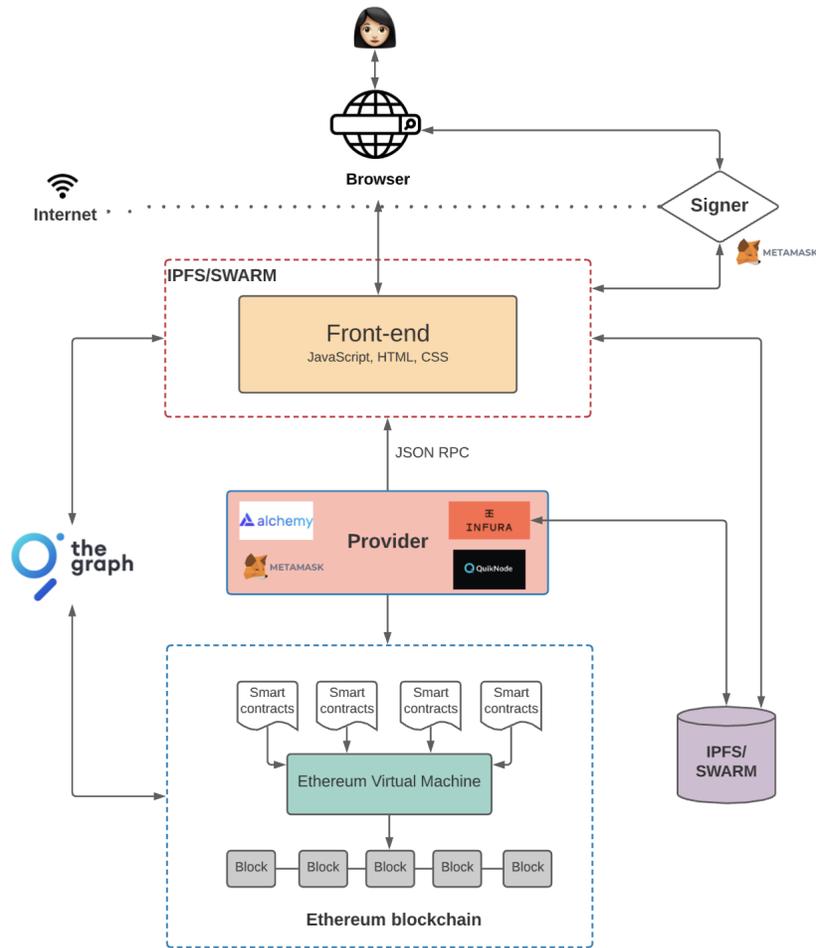


Figura 2.5 Estructura de aplicaciones de Web2⁷

⁷ <https://www.preethikasireddy.com/post/the-architecture-of-a-web-3-0-application>

2.2.2. Conceptos importantes dentro de la Blockchain

- **Prueba de trabajo**

La prueba de trabajo (del inglés *Proof of Work*, PoW), como indican en el libro *Mastering Ethereum: building smart contracts and DApps*[13], “Es una pieza de datos (la prueba) que requiere un cálculo significativo para encontrar”. Se trata del protocolo de consenso más conocido que consiste en que las partes de una red realicen con éxito un trabajo computacionalmente costoso para acceder a los recursos de dicha red. Esto provoca ineficiencia, pues existen alternativas como la prueba de participación que consiguen lo mismo de manera más eficiente.

- **Prueba de participación**

La prueba de participación (del inglés *Proof of Stake*, PoS), es un protocolo de consenso que reemplaza al protocolo del punto anterior aportando seguridad, escalabilidad a la red y mayor eficiencia. Existen unos participantes de la red llamados validadores que acceden a los recursos de dicha red a cambio de tener bloqueada una cierta cantidad de criptomonedas.

Se comenta en el libro *Mastering Ethereum: building smart contracts and DApps*[13]:

“PoS pide a los usuarios que demuestren la propiedad de una cierta cantidad de criptomoneda (su ‘participación’ en la red) para poder participar en la validación de las transacciones.”

No requiere trabajos computacionalmente costosos, por lo que es más eficiente que la prueba de trabajo.

- **Criptomonedas**

Como monedas físicas se usan para facilitar transacciones financieras, las criptomonedas hacen la misma función pero siendo digitales, se usan tanto para transacciones financieras como contratos inteligentes en Internet de manera descentralizada, utilizando criptografía y programaciones avanzadas.

- **Monedero digital**

A diferencia del monedero físico, el monedero digital no contiene criptomonedas, sino usa las claves criptográficas necesarias para acceder a las criptomonedas en la red

correspondiente. Esto significa que el monedero digital es más bien una llave electrónica que permite las acciones de comprar, vender y mantener las criptomonedas de forma segura y accesible.

En este proyecto se ha utilizado MetaMask⁸ como el monedero digital, ya que es muy potente y dispone de extensión de Chrome⁹.

- **Token**

Se llama token a una unidad de valor basada en Blockchain y emitida por un contrato inteligente. Un token no es una criptomoneda, aunque muchas veces se confunden estos dos términos pensando que son lo mismo. Los tokens sirven para múltiples usos como el pago de comisiones dentro de una aplicación descentralizada basada en Blockchain o para votar propuesta en una aplicación descentralizada autónoma.

- **Contrato inteligente**

Como indican en el capítulo 2 del libro *Blockchain: blueprint for a new economy*[11],

“En el contexto de la Blockchain, los contratos inteligentes significan transacciones de cadena de bloques que van más allá de las simples transacciones de compra/venta de divisas y pueden tener instrucciones más detalladas integradas.”

Para entender qué es el contrato inteligente, se recuerda lo que es un contrato, según el capítulo 2 del libro *Blockchain: blueprint for a new economy*[11]:

“Un contrato en el sentido tradicional es un acuerdo entre dos o más partes para hacer o no hacer algo a cambio de otra cosa.”

Los contratos hasta ahora han sido documentos escritos, que requieren muchos costes, tiempo y terceros que intervienen en el proceso. Lo peor, los contenidos de estos contratos pueden estar mal interpretados.

En cambio, los contratos inteligentes son unos “*scripts*” (códigos informáticos) escritos con lenguajes de programación que se pueden ejecutar automáticamente. Por lo cual, se puede evitar el lastre de la interpretación al no ser escrito en los lenguajes que hablamos.

⁸ <https://metamask.io/>

⁹ https://www.google.com/intl/es_es/chrome/

Además, el contrato inteligente se puede crear por personas físicas y jurídicas, o por máquinas y programas, y su contenido es visible por todos y que no se puede cambiar al existir sobre la tecnología blockchain. Esto le confiere un carácter descentralizado, inmutable y transparente.

2.2.3. Ventajas de la Blockchain

Las principales ventajas de la Blockchain son:

- **Descentralización:** Al realizar la aprobación de transacciones de forma descentralizada, se evita un poder centralizado que tome decisiones que afectan al resto y consigue una mayor democracia entre los integrantes de la red.
- **Inmutabilidad:** Como indican en el documento *The Advantages and Disadvantages of the Blockchain Technology*[14]:

“Lo inmutable se logra en las transacciones acordadas y compartidas a través de Blockchain. Cuando la transacción se conecta a Blockchain, no es posible cambiarla o eliminarla.”

Si bien puede ser vista como desventaja ante errores que podrían no ser subsanados, la inmutabilidad que ofrece Blockchain permite que las transacciones se lleven a cabo sin alteración posible.

- **Transparencia:** Una Blockchain pública como Ethereum, permite ver todas sus transacciones realizadas de forma pública, lo cual posibilita la trazabilidad pero siempre de forma anónima.
- **Red distribuida:** Nadie es propietario de la red y todos los integrantes de la red guardan la misma copia de datos de la red, por lo que si un integrante sufre un percance, no existiría ningún problema a diferencia de un sistema no distribuido.
- **Bajos costes para usuarios:** Las transacciones, por lo general, conllevan un coste mucho mejor que en la banca tradicional.
- **Rapidez:** Las transacciones en la Blockchain tardan muy poco en comparación a la banca tradicional.
- **Seguridad:** Como indican en el libro *Blockchain: blueprint for a new economy*[11]:

“Una de las principales ventajas es que Blockchain es una tecnología de empujar (el usuario inicia y envía información relevante al red solo para esta transacción),

no es una tecnología de tirar (como para una tarjeta de crédito o banco, la información personal del usuario está archivada para ser extraída cada vez que se autorice).”

De esta manera, se evita que los datos del usuario estén archivados en varios sitios y garantiza la mayor seguridad.

2.2.4. Desventajas de la Blockchain

Las principales desventajas de la Blockchain son:

- **Alto consumo de energía:** Como indican en el documento *The Advantages and Disadvantages of the Blockchain Technology*[14], “La principal desventaja de Blockchain es el alto consumo de energía.”

Algunas redes Blockchain utilizan el protocolo de consenso prueba de trabajo, el cual consume demasiada energía.

Ethereum planea cambiar este por el protocolo de consenso prueba de participación, que al tener un consumo ya razonable, provocaría que dejara de existir esta desventaja.

- **Sin copia de seguridad:** Como indican en la introducción del libro *Blockchain: blueprint for a new economy*[11], si un usuario pierde su clave privada para acceder a la Blockchain, no hay manera para recuperar su cuenta y dinero.

Capítulo 3 - Tecnologías utilizadas

3.1. Javascript

Javascript¹⁰ es un lenguaje de programación que se utiliza como complemento de HTML y CSS para crear páginas web debido a su integración nativa en los navegadores (lenguaje interpretado sin necesidad de compilación). También es utilizado en la actualidad por muchos marcos de trabajo (del inglés *frameworks*) que permiten el desarrollo de aplicaciones de diferentes tipos.

En este proyecto, se trata del lenguaje utilizado tanto en front-end como en back-end.

3.2. React

Como indican en el libro *Learning React: Functional Web Development with React and Redux*[9]: “React es una biblioteca popular utilizada para crear interfaces de usuario.” React¹¹ es un framework bajo entorno Node.js (entorno de ejecución para aplicaciones JavaScript) que utiliza el lenguaje Javascript para ayudar a crear front-end interactivas de forma sencilla. Se encarga de actualizar y renderizar los componentes de la página web de manera eficiente.

Se trata del framework utilizado en este proyecto para el front-end.

3.3. GitHub

Github¹² es una forja para proyectos software utilizando si sistema de control de versiones GIT¹³ . Es muy conocida por los desarrolladores y utilizada por los autores de este proyecto durante el estudio de la carrera.

Se ha utilizado en este proyecto tanto para alojar el código del front-end como del back-end.

¹⁰ <https://developer.mozilla.org/es/docs/Web/JavaScript>

¹¹ <https://es.reactjs.org/>

¹² <https://github.com/>

¹³ <https://git-scm.com/>

3.4. WebStorm

WebStorm¹⁴ es un entorno de desarrollo integrado (de inglés *Integrated Development Environment*, IDE) para JavaScript y tecnologías relacionadas (ver la [Figura 3.1](#)). Como indican en su página oficial: “WebStorm es el IDE más inteligente para JavaScript”. Tiene una interfaz muy intuitiva y hace que el proceso de desarrollo sea más agradable y organizado.

Es el IDE elegido para el desarrollo de este proyecto.

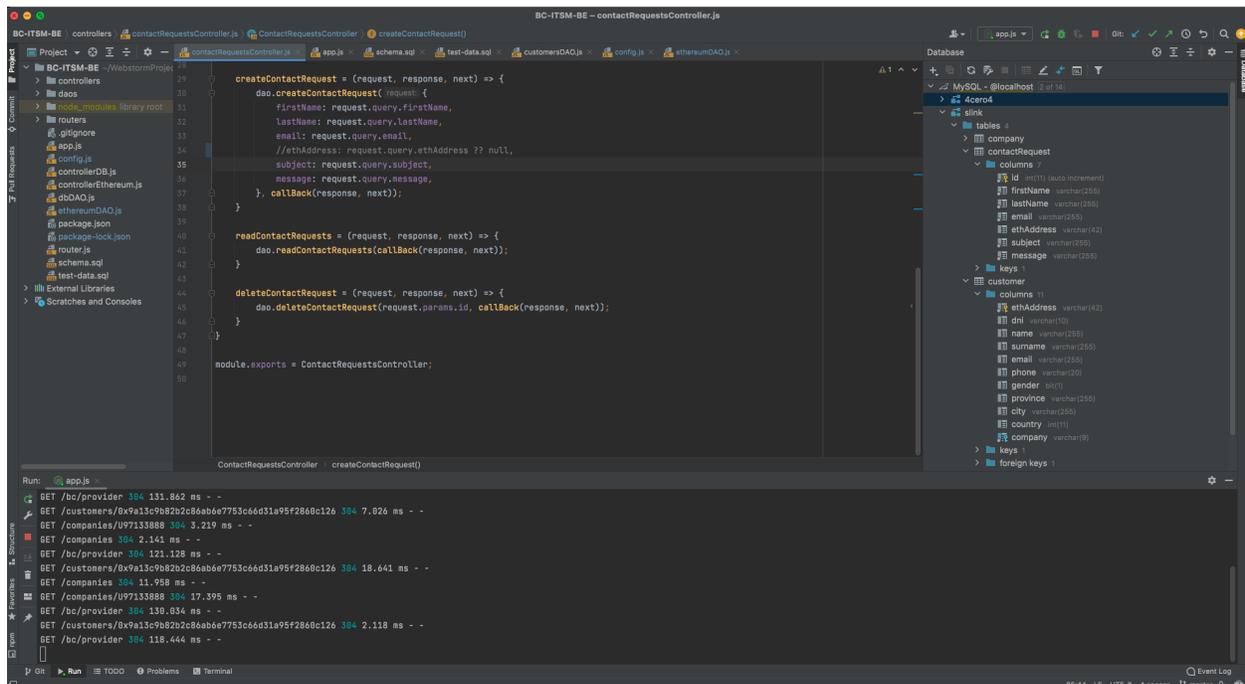


Figura 3.1 WebStorm

3.5. Postman

Postman¹⁵ es una aplicación que ejerce de cliente HTTP para permitir testear peticiones HTTP request (del inglés, *HTTP request*) a través de una interfaz gráfica de usuario.

Es una herramienta utilizada en el proyecto para el desarrollo de la Interfaz de Programación de Aplicaciones (del inglés *Application Programming Interface*, API) con back-end y The Graph.

¹⁴ <https://www.jetbrains.com/es-es/webstorm/>

¹⁵ <https://www.postman.com/>

En la [Figura 3.2](#) se muestra cómo realizar la petición de obtención de clientes a través de Postman, en la sección [4.5.3. API REST](#) se explicarán todas las peticiones HTTP realizadas en el back-end de este proyecto.

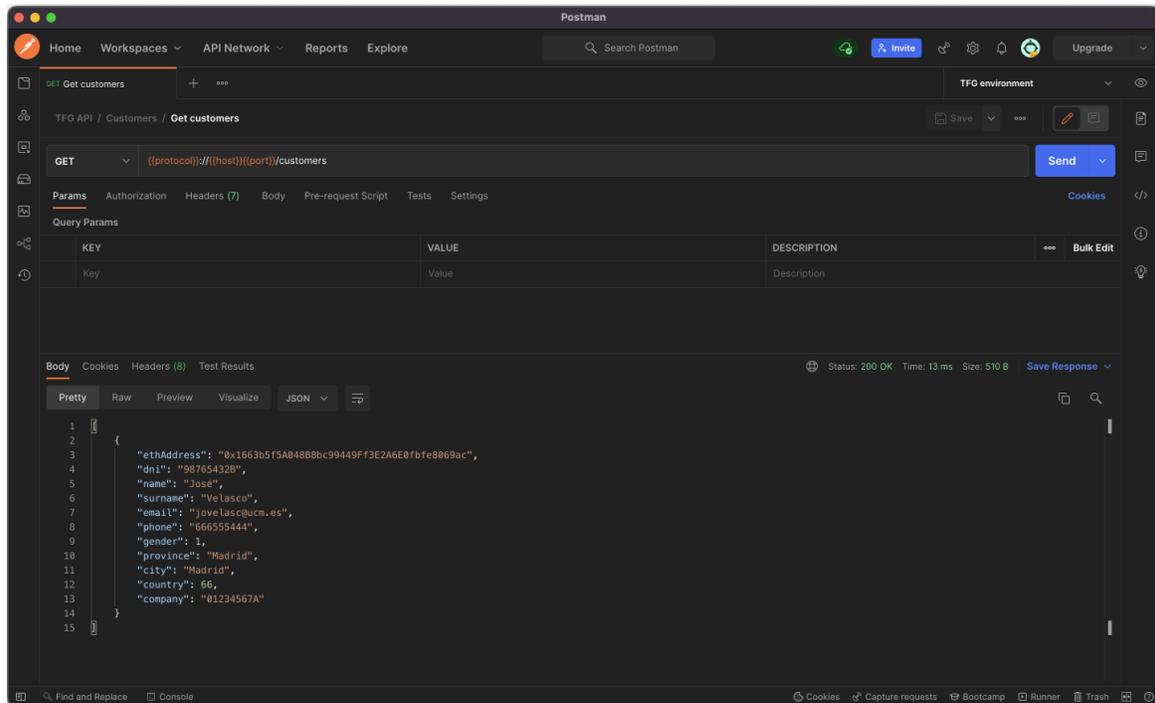


Figura 3.2 Postman

3.6. Ethereum

3.6.1. ¿Qué es Ethereum?

Ethereum¹⁶ es una plataforma digital que adopta la tecnología Blockchain y expande su uso a una gran variedad de aplicaciones. A día de hoy, se trata de la red Blockchain con contratos inteligentes donde más desarrollo existe, a pesar de sus altas comisiones y lentitud de transacciones en la actualidad frente a otras redes Blockchain. Prueba de ello, es su capitalización de mercado, la mayor de todas las redes Blockchain después de Bitcoin.

¹⁶ <https://ethereum.org/en/whitepaper/>

3.6.2. ¿Cómo funciona Ethereum?

En la red de Ethereum, existe un ordenador único y canónico (llamado máquina virtual de Ethereum o EVM), cuyo estado han acordado todos los participantes de la red. Cada usuario (nodo) de la red de Ethereum mantiene una copia del estado de EVM, y cuando un usuario quiere emitir una solicitud de transacción (un cambio de estado en la EVM), todos los demás la verifican, validan.

El estado actual de la EVM se almacena en la Blockchain que, a su vez, almacenan y acuerdan todos los usuarios. De esta forma, se garantiza la protección en la red Ethereum al máximo.

Al igual que Bitcoin (la red Blockchain pionera), se trata de una red Blockchain con protocolo de consenso prueba de trabajo (explicado en la sección [2.2.2. Conceptos importantes dentro de la Blockchain](#)) lo que implica un gasto energético superior a otras redes Blockchain que utilizan el protocolo de consenso prueba de participación (explicado al principio de este apartado). También implica un gasto en comisiones superior por la misma razón.

Aún así, se trata de la red Blockchain más utilizada para desarrollo de aplicaciones de propósito general, siendo prueba de ello la capitalización de mercado actual de Ethereum y la cantidad de proyectos relacionados.

La criptomoneda utilizada en esta red es el Ether (ETH), utilizada tanto para pagar las transacciones como para envío de dinero entre direcciones.

Por otro lado, mientras que Bitcoin es la red pionera de Blockchain con el objetivo de ser un sistema de pagos financieros, Ethereum tiene el objetivo de crear aplicaciones de propósito general, existiendo ya muchas de ellas.

3.6.3. Transacciones en Ethereum

Como indican en *Ethereum White Paper*[7], una transacción en Ethereum es “*un paquete de datos firmado que almacena un mensaje que se enviará desde una cuenta de propiedad externa*”. Las transacciones contienen:

1. El destinatario de mensaje
2. Una firma que identifique al remitente
3. La cantidad de ETH a transferir del remitente al destinatario
4. Datos opcionales

5. Un valor STARTGAS, que representa el número máximo de pasos computacionales que la ejecución de la transacción puede tomar.
6. Un valor de GASPRICE, que representa la tarifa que paga el remitente por paso computacional.

3.6.4. Estructura de Ethereum

Ethereum se puede dividir por niveles¹⁷:

1. **Máquina virtual de Ethereum:** Entorno para contratos inteligentes en Ethereum.
2. **Contratos inteligentes:** Programas ejecutables en la red Blockchain de Ethereum.
3. **Nodos de Ethereum:** Conocidos como mineros, forman la red Blockchain, validan transacciones y permiten interactuar con la red.
4. **APIs de clientes Ethereum:** Librerías y aplicaciones que se comunican con la red de Ethereum.
5. **Aplicaciones de usuario final.**

Es la red utiliza el proyecto para almacenar y recuperar SLAs en la Blockchain.

3.7. Remix

Remix¹⁸ es un entorno de desarrollo de Ethereum para compilar y desplegar contratos inteligentes a la Blockchain. Es el IDE que se utiliza para realizar pruebas de contratos inteligentes y desplegarlos en entorno real en el proyecto.

La [Figura 3.3](#) muestra un ejemplo de despliegue en Remix. Se puede conocer más sobre el despliegue en Remix en este proyecto en la sección [4.6.1. Despliegue del contrato inteligente](#).

¹⁷ <https://ethereum.org/en/developers/docs/ethereum-stack/>

¹⁸ <https://remix-project.org/>

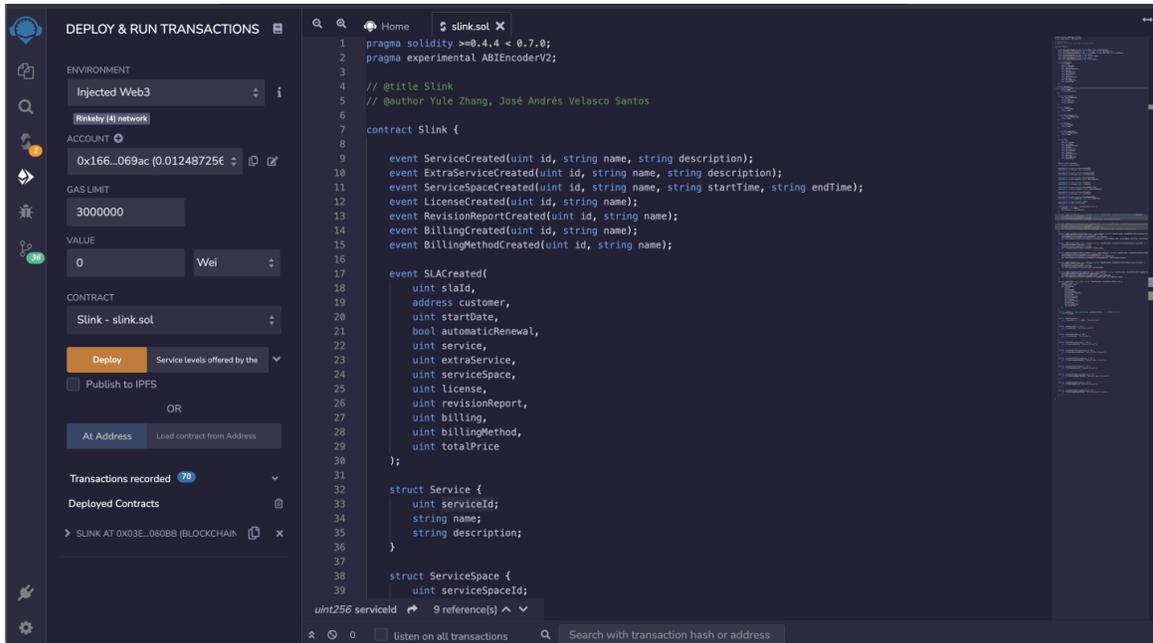


Figura 3.3 Remix

3.8. Solidity

Solidity¹⁹ es el lenguaje con sintaxis parecida a JavaScript, utilizado para crear contratos inteligentes y enviarlos a la red Ethereum. Se puede consultar la sección [4.4. Contrato inteligente](#) para ver el contrato inteligente de este proyecto.

3.9. SQL

SQL²⁰ es el lenguaje de consultas sobre bases de datos relacionales utilizado en este proyecto para trabajar con los datos sensibles de los SLAs en una base de datos relacional privada.

3.10. MySQL

MySQL²¹ es un sistema de bases de datos relacional, que permite crear y manejar bases de datos basadas en tablas.

Se utiliza para guardar los datos sensibles de los clientes de los SLAs en este proyecto.

¹⁹ <https://solidity-es.readthedocs.io/es/latest/>

²⁰ <https://es.wikipedia.org/wiki/SQL>

²¹ <https://www.mysql.com/>

3.11. Express

Express²² es un framework de back-end bajo entorno Node.js, diseñado para crear aplicaciones web y APIs REST. Permite definir rutas de métodos HTTP como peticiones GET, POST, PUT, y DELETE.

En este proyecto se utiliza principalmente para guardar los datos sensibles de los SLAs, aunque también sirve de puente para realizar consultas sobre la Blockchain con The Graph (ver la sección [3.13. The Graph](#)).

3.12. GraphQL

GraphQL²³ es un lenguaje de consulta y manipulación de datos para APIs que, a diferencia de una API REST, permite definir a los clientes la estructura de datos requerida de forma flexible y sencilla.

3.13. The Graph

3.13.1. ¿Qué es *The Graph*?

Como indican en la página de The Graph²⁴:

“The Graph es un protocolo de indexación que permite consultar y procesar datos de redes como Ethereum. Cualquier persona puede crear y publicar una API pública, denominados como subgrafos, facilitando el acceso a la información correspondiente.”

The Graph es un proyecto utilizado ya en muchos otros proyectos famosos, como SushiSwap (sistema de intercambio de criptomonedas descentralizado), Uniswap (similar al anterior), Decentraland (plataforma Blockchain y de realidad virtual) y Aragon (framework centrado en facilitar la creación de Organizaciones Autónomas Descentralizadas, DAO).

Se trata de un protocolo lo suficientemente maduro para ser utilizado en este proyecto.

²² <https://expressjs.com/es/>

²³ <https://graphql.org/>

²⁴ <https://thegraph.com/es/>

3.13.2. ¿Cómo funciona The Graph?

Los contratos inteligentes de Ethereum permiten emitir eventos personalizados en el momento deseado. Los eventos son un mecanismo que proporciona de manera nativa la máquina virtual de Ethereum para registrar información que es visible para cualquier interesado externo. Es la manera en la cual se informa de un hecho ocurrido en un contrato inteligente.

The Graph, gracias a subgrafos (explicado a continuación), utiliza estos eventos para indexar la información proporcionada por estos.

En la red de The Graph, existen nodos de grafos encargados de escanear continuamente la red Ethereum en busca de nuevos bloques y los datos de subgrafos. Una vez encuentran datos nuevos, se formatean según lo indicado en el subgrafo correspondiente para ser consultados posteriormente.

Una vez se han recopilado los datos, pueden ser consultados de forma simple y flexible vía API con GraphQL.

3.13.3. ¿Qué es un subgrafo?

Un subgrafo es una configuración para obtención de datos por parte de nodos de grafos, que permite no solo indicar la dirección del contrato en Ethereum, sino también qué datos son recogidos, formateo de estos y estructura de datos para ser consultados posteriormente con GraphQL.

3.13.4. ¿Qué funciones existen dentro de la red?

La red está formada por 4 roles:

- Delegador: encargado de proteger la red al delegar los tokens de The Graph a indexadores.
- Curador: organizador de datos mediante la señalización de subgrafos.
- Indexador: operador de nodos encargado de indexar los datos y proveer consultas.
- Desarrollador: creador de subgrafos o consumidor de otros ya existentes.

El último rol es el ejercido en este proyecto.

Capítulo 4 - Sistema descentralizado basado en Blockchain que almacena SLAs, SLink

En este capítulo se expondrá el sistema SLink que se ha desarrollado para este proyecto. SLink es un sistema descentralizado basado en la tecnología Blockchain, que favorece la gestión de los SLAs que se establecen entre los proveedores de servicios de TI y sus clientes siguiendo las buenas prácticas de ITIL.

4.1. Nombre

Se ha dado el nombre “*SLink*” al sistema, las dos primeras letras “*SL*” representan a “*SLA*” y “*Link*” es una palabra inglesa que significa “*conexión*” en español. La idea de este nombre del sistema viene de su función, que es establecer conexión con la Blockchain para almacenar y recuperar los SLAs.

4.2. Arquitectura

La arquitectura de la aplicación consta de varios apartados que interaccionan entre ellos como se explica a lo largo de este capítulo. La estructura del sistema está compuesta por:

- **Base de datos privada:** El sistema utiliza MySQL como sistema de gestión de bases de datos relacionales para almacenar los datos sensibles de los SLAs.
- **Back-end:** Express es la tecnología que media entre front-end, The Graph y la base de datos privada. Se encarga de proporcionar al front-end acceso a datos tanto públicos como privados y de registrar datos públicos proporcionados por el front-end.
- **Front-end:** React proporciona la interfaz visible de SLink a los usuarios, permitiendo que estos puedan crear SLAs y en caso del administrador, gestionar usuarios, empresas y ver el tablero de KPIs.
- **Monedero digital:** Se trata del medio que permite al sistema firmar transacciones en la red Blockchain de Ethereum. Es necesario para firmar cambios en el contrato inteligente del sistema que permite registrar un nuevo SLA. Hay muchos tipos de monederos digitales para elegir, se ha utilizado MetaMask en este proyecto.

- **Blockchain:** Ethereum es la blockchain que utiliza el sistema para guardar de forma pública, segura y permanente los SLAs creados. Contiene los contratos inteligentes del sistema.
- **The Graph:** Es el motor de búsqueda de datos sobre los contratos inteligentes que utiliza el sistema para obtener los datos necesarios.

La [Figura 4.1](#) ilustra la arquitectura general del sistema implementado.

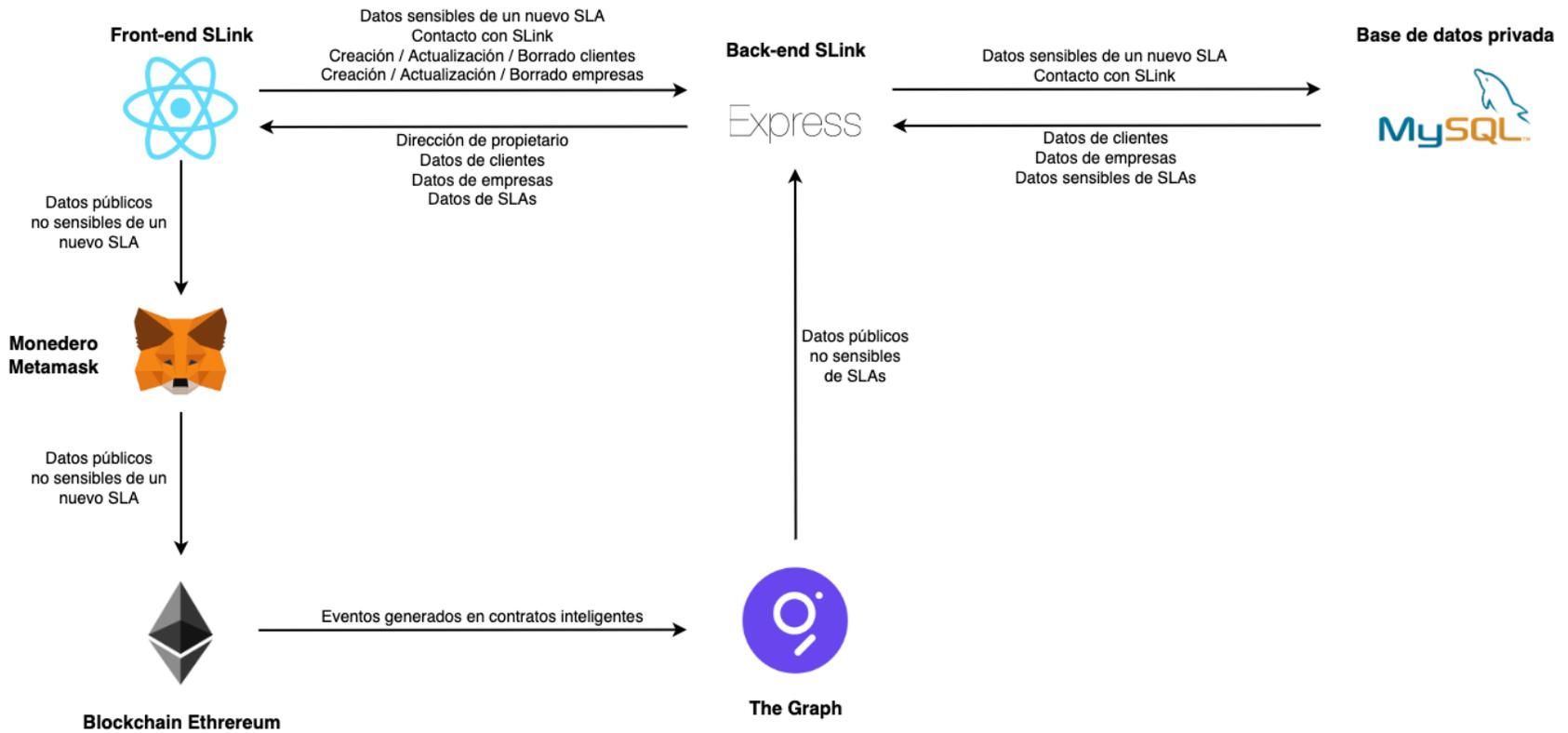


Figura 4.1 Diagrama de arquitectura del sistema

4.3. Front-end

En esta sección se expondrán todas las tecnologías y funcionalidades que se pueden observar en el front-end del sistema SLink.

4.3.1. *Página principal*

En la [Figura 4.2](#) se puede observar la página principal del sistema SLink, que consta de un diagrama giratorio de 360 grados con nodos y líneas azules en el fondo, cada nodo está conectado a través de las líneas con el resto. De esta manera, se asimila a la red de Blockchain, donde cada nodo tiene el mismo registro de datos que el resto de nodos.

En la parte superior de la página está el nombre del sistema, SLink, y la licencia elegida para este sistema (ver la sección [4.7.1. Licencia](#) para conocer más detalle sobre la licencia).

Se ha aplicado un efecto de teclado a la frase *“Abrir tu SLA AHORA, Abrir tu SLA en BlockChain”* en color blanco, situada debajo del diagrama giratorio para que la interfaz sea más intuitiva y llamativa.

Se disponen de dos botones en la parte inferior de la página, si lo que se quiere es enviar y registrar un SLA, al clicar el botón izquierdo, se llevará a la página de SLA (ver la sección [4.3.2. Página de SLA y elementos de SLA en SLink](#)). Y si lo que se quiere es conocer sobre SLink y enviar consulta al sistema, al clicar el derecho, se llevará a la página de información y contactos (ver la sección [4.3.9. Página de Conocernos](#)).



Figura 4.2 Página principal

4.3.2. Página de SLA y elementos de SLA en SLink

Antes de conocer la página de SLA, se procede a explicar primero los elementos de SLA aplicados para el sistema SLink, se verán estos elementos en la página de SLA.

Anteriormente, en la sección [2.1.5. Introducción a los SLAs](#), se hablaba de los elementos que pueden tener los SLAs, y en este sentido, hay que tener cuenta de que los elementos de los SLAs se podrían y se deberían modificar según las necesidades y funcionalidades de cada organización. En este caso SLink siendo un sistema creado con el fin de estudio para un Trabajo Fin de Grado (TFG), se ha adaptado una versión estándar incluyendo los elementos más relevantes y comunes de un SLA (ver la [Tabla 4.1](#) para conocer más en detalle).

Atributos	Contenido
Partes contratantes	Datos de los participantes en el acuerdo: cliente (Dirección Ethereum, DNI, nombre y apellidos, sexo, email, teléfono, nombre de empresa, dirección de empresa, código fiscal de empresa), proveedor (nombre de empresa, dirección de empresa, código fiscal de empresa).
Duración del SLA	Periodo de vigencia del acuerdo según las partes contratantes (fecha de inicio y fin).

Atributos	Contenido
Servicios cubiertos	Servicios prestados por el proveedor al cliente.
Horario de servicio	Horarios en los que se prestará el servicio (hora de inicio y fin).
Licencias	Posibles licencias que se necesitarán durante el curso de servicio.
Niveles de servicio	Parámetros bajo los que se prestará el servicio.
Facturación	Periodos y métodos de facturación.

Tabla 4.1 Elementos de SLA en SLink

La página de SLA se trata de un formulario donde se recogen los datos que se indican en la [Tabla 4.1](#). Por motivos de seguridad, para poder visualizar correctamente dicha página, se necesita que el cliente ya se haya dado de alta en el sistema junto con su dirección Ethereum. La manera de darse de alta en el sistema es a través de hacer clic en el botón “*Conocenos*” en la página principal (ver la [Figura 4.2](#)) o el link “*contacte con nosotros*” en la [Figura 4.3](#), en ambos casos se llegará a la página de información y contactos (ver la sección [4.3.9. Página de Conocernos](#)) para poder contactar con nosotros.

Si una persona no es cliente del sistema, es decir, después de conectar con su monedero digital (ver la sección [4.3.3. Conectar con el monedero digital](#) para conocer más detalle sobre cómo conectar con el monedero digital), aparece que la dirección de Ethereum no está registrada en el sistema, se verá la pantalla que se muestra en la [Figura 4.3](#) con la información mostrada en la [Figura 4.4](#).

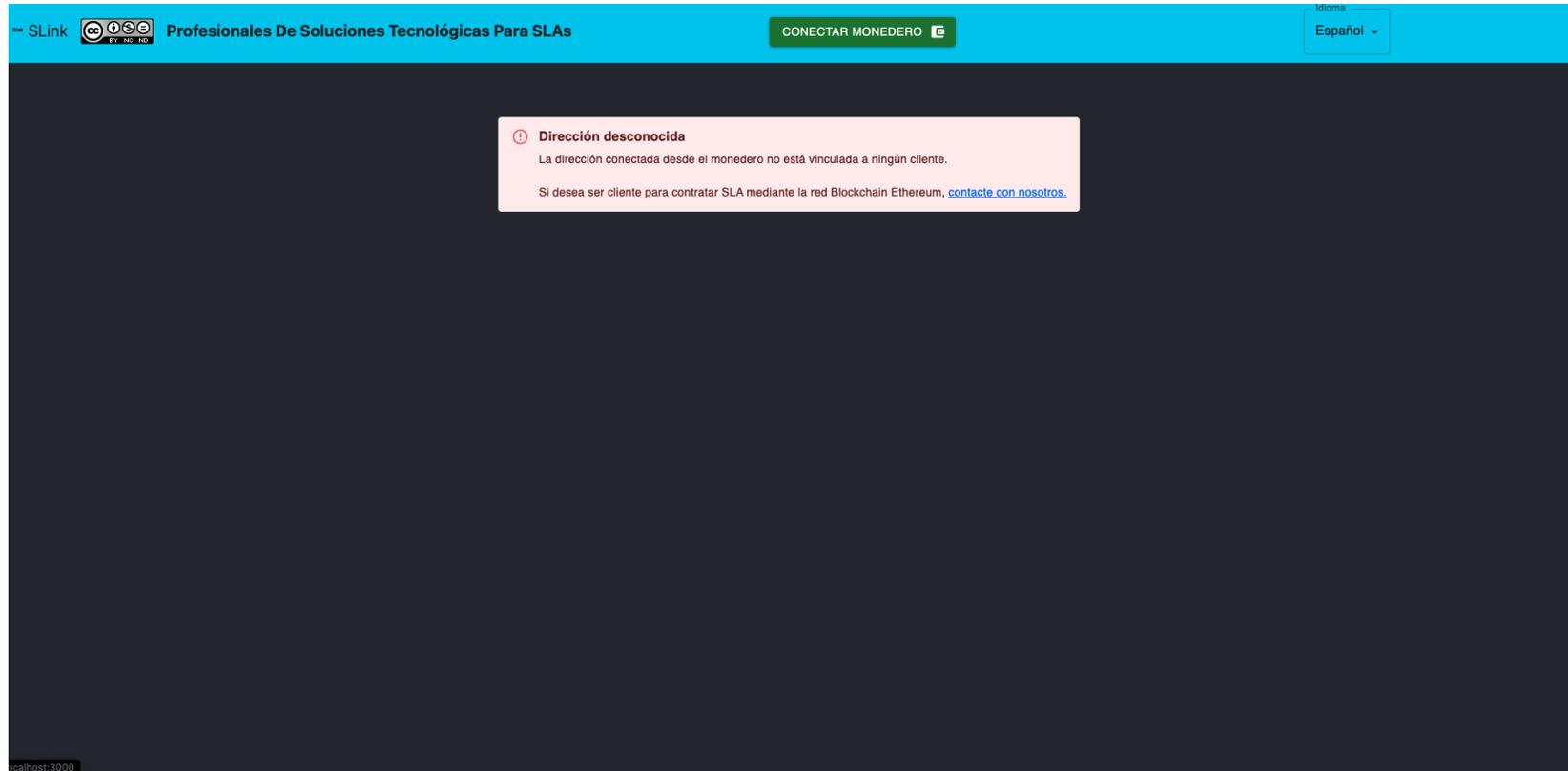


Figura 4.3 Página de SLA (visitante)



Dirección desconocida

La dirección conectada desde el monedero no está vinculada a ningún cliente.

Si desea ser cliente para contratar SLA mediante la red Blockchain Ethereum, [contacte con nosotros.](#)

Figura 4.4 Información de error (visitante)

En cambio, si una persona es cliente del sistema, se verán las pantallas mostradas en la [Figura 4.5](#) y la [Figura 4.6](#)) organizadas en forma de formulario. Este formulario tiene 3 partes separadas que abarcan toda la información indicada en [Tabla 4.1](#):

1. **Información del cliente:** Son campos no editables, ya que esta parte de información está almacenada en la base de datos privada cuando el cliente se registró en el sistema. Al entrar con su propia dirección de Ethereum, se recupera la información de la base de datos. La información del cliente disponible en esta parte son:
 - Dirección de Ethereum
 - DNI
 - Nombre y apellidos
 - Sexo
 - Email
 - Teléfono
 - CIF y nombre de empresa
 - Dirección de empresa
2. **Información de la empresa:** Igual que la parte de información del cliente, los campos de esta parte también son no editables y que se recupere de la base de datos. La información de la empresa disponible son:
 - CIF y nombre de empresa
 - Dirección de empresa
3. **Detalles SLA:** Se pedirá la siguiente información de SLA al cliente para poder enviarlo:
 - Fecha inicial y si la renovación automática
 - Horario y cobertura de servicio
 - Soporte extra
 - Pagador de licencias

- Nivel de servicio
- Periodo de reporte
- Periodo y método de facturación

Cabe destacar que al final de esta página se dispone de un campo de cálculo automático del precio total de SLA, hay que tener en cuenta que este precio es de referencia, el precio real se puede variar según las necesidades de cada cliente.

SLink  Profesionales De Soluciones Tecnológicas Para SLAs LISTA ENTIDADES DASHBOARD DESCONECTAR 0X9A1...C126 Idioma Español

 **Dirección conocida**
Hola de nuevo, Yule.

Información del cliente

Dirección ethereum: 0x9a13C9B82b2C86AB6e7753C86d31a95f2860c126

DNI: Y7166898C

Nombre: Yule

Apellidos: Zhang

Seleccionar género ...: Femenino

Email: yulzhang@ucm.es

Teléfono: 555444333

Empresa: U97133888 (Fdl - UCM)

Provincia: Madrid

Ciudad: Madrid

País: Andorra

Información de la empresa

CIF: U97133888

Nombre empresa: Fdl - UCM

Dirección empresa: Ciudad Universitaria Madrid

Detalles SLA

Fecha inicial: 

Renovación automática: Sí No

Figura 4.5 Página de SLA (cliente) - parte 1

SLink  **Profesionales De Soluciones Tecnológicas Para SLAs** [LISTA ENTIDADES](#) [DASHBOARD](#) [DESCONECTAR 0X166...69AC](#) Idioma Español

Fecha inicial Renovación automática

Sí No

Horario de servicio
 Cobertura de servicios

Soporte extra
 Pagador de licencias

Niveles de servicio

Niveles de servicio ofertados por la empresa:

Reporte de revisión

Período de reporte

Facturación de servicios

Período de facturación
 Método de facturación

He leído y acepto la política de privacidad
 Leer el [aviso legal y la política de privacidad](#).

El precio actual de tu SLA es: 0€/year

Los campos con * son obligatorios en este formulario. El precio mostrado anteriormente no está considerando el coste del servicio extra, ya que se cobrará diariamente de acuerdo a la cantidad de solicitud.

Figura 4.6 Página de SLA (cliente) - parte 2

4.3.3. Conectar con el monedero digital

Para conectar con el monedero digital, simplemente hay que hacer clic en el botón verde “Conectar monedero” colocado en la parte superior de cada página (excepto la principal, tal y como se muestra en la [Figura 4.7](#)). Al hacer clic en el botón, se muestra una ventana emergente con todos los monederos disponibles (ver la [Figura 4.8](#)), en este caso, se ha utilizado la extensión de MetaMask²⁵ en Chrome, la cual se puede descargar a través de la tienda online de Chrome. MetaMask hace de puente entre el sistema SLink y la red Ethereum.

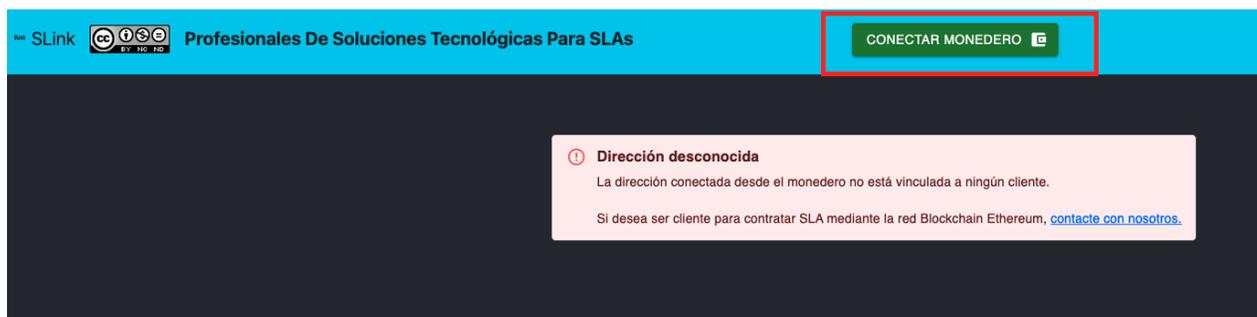


Figura 4.7 Botón Conectar monedero

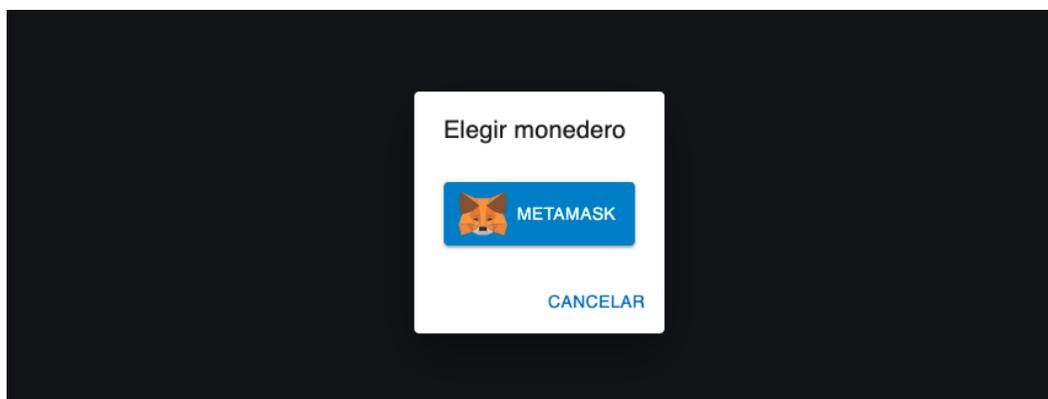


Figura 4.8 Elegir monedero

Después de haber seleccionado MetaMask, se pide introducir la contraseña de la cuenta, tal y como se muestra en la [Figura 4.9](#).

²⁵

<https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaoehlefnkodbefgpgknn>

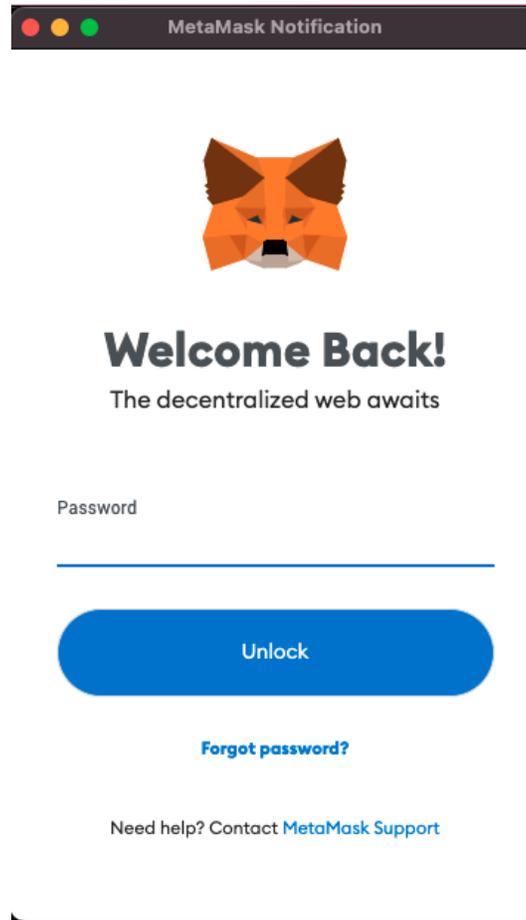


Figura 4.9 Notificación MetaMask

A continuación, hay que seleccionar la cuenta de Ethereum que se quiere conectar (ver la [Figura 4.10](#)). Al estar conectado con la cuenta de Ethereum, si dicha cuenta ya está registrada en el sistema, se verá la página de SLA correctamente (ver la [Figura 4.5](#)). En cambio, si no se ha conectado se verá el error que se muestra en la [Figura 4.3](#).

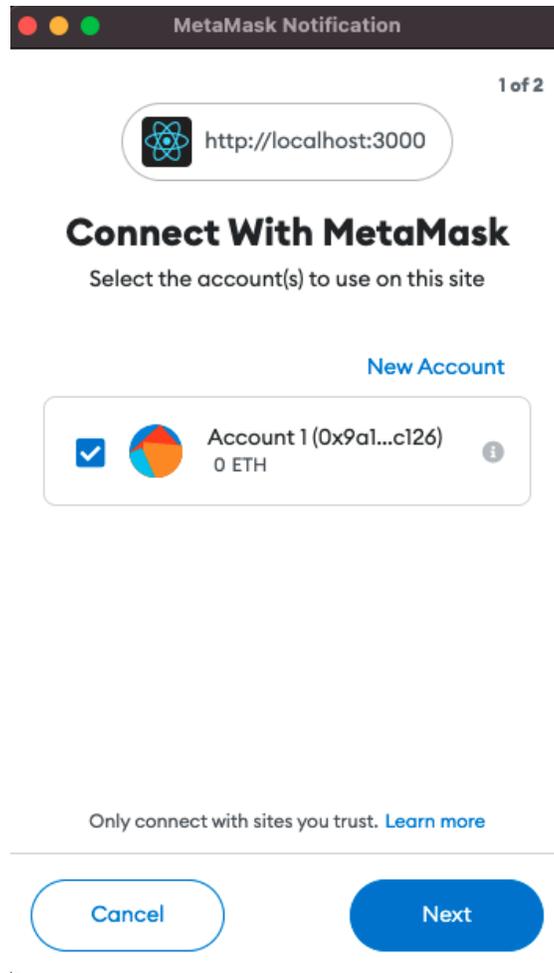


Figura 4.10 Selección de cuenta Ethereum en MetaMask

4.3.4. Creación de un SLA

Una vez se tiene conectado el monedero digital, se puede proceder a la creación de un SLA, seleccionando las opciones deseadas, tal y como se muestra en la [Figura 4.5](#) y en la [Figura 4.6](#).

Para crear el SLA, se debe hacer clic en el botón “*Enviar*” en la parte inferior izquierda. Esto llevará a pedir confirmación del coste de las tasas de la transacción en MetaMask, como se puede ver en la [Figura 4.11](#).

Una vez aceptadas dichas tasas, la transacción se iniciará en la red Blockchain de Ethereum. Mientras está en marcha, el sistema muestra con un fondo amarillo el identificador del SLA creado, el identificador de la transacción en marcha (incluido un enlace para ver detalles de la

transacción en tiempo real desde Etherscan) y un enlace para crear un nuevo SLA (como se puede ver en la [Figura 4.12](#)).

Cuando la transacción finaliza, el sistema muestra la misma información pero ahora con un fondo verde y un mensaje que indica que la transacción ha finalizado (como se puede ver en la [Figura 4.13](#)).

Etherscan²⁶ es una página web que permite ver las transacciones de Ethereum y, entre otras muchas cosas, visualizar el código de muchos contratos inteligentes.

²⁶ <https://etherscan.io/>

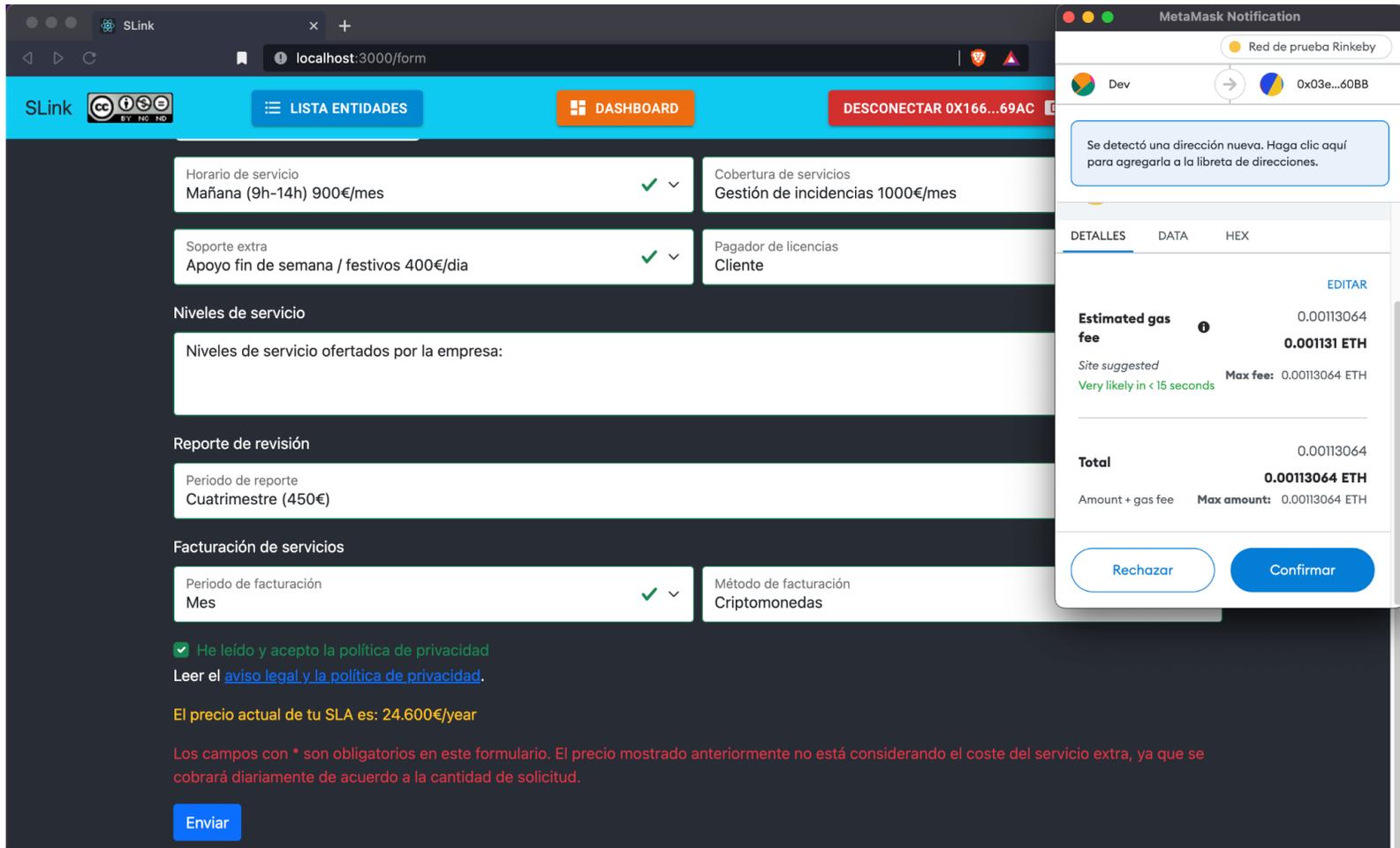


Figura 4.11 Confirmación de pago de tasas de transacción Ethereum - paso 1

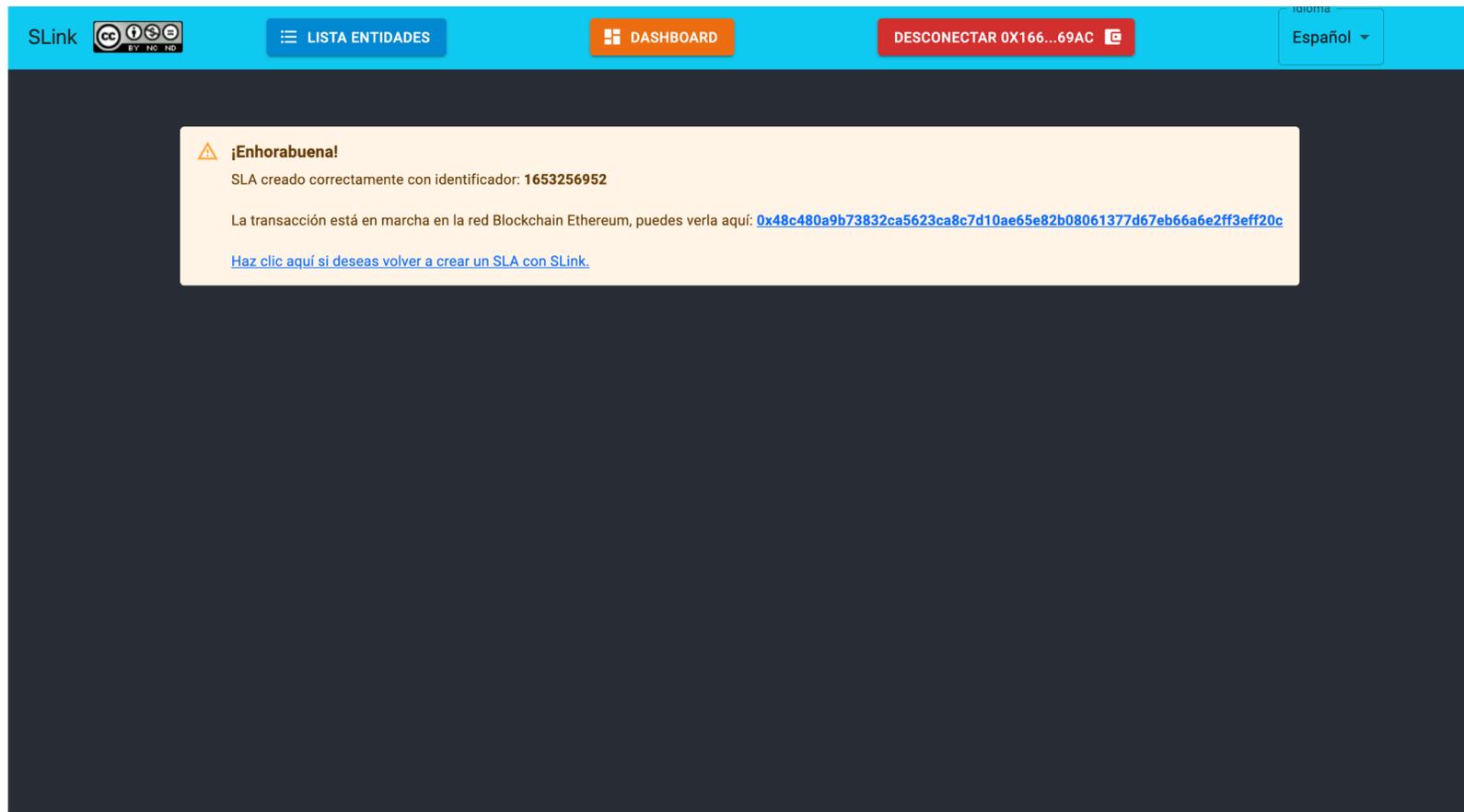


Figura 4.12 Confirmación de pago de tasas de transacción Ethereum - paso 2

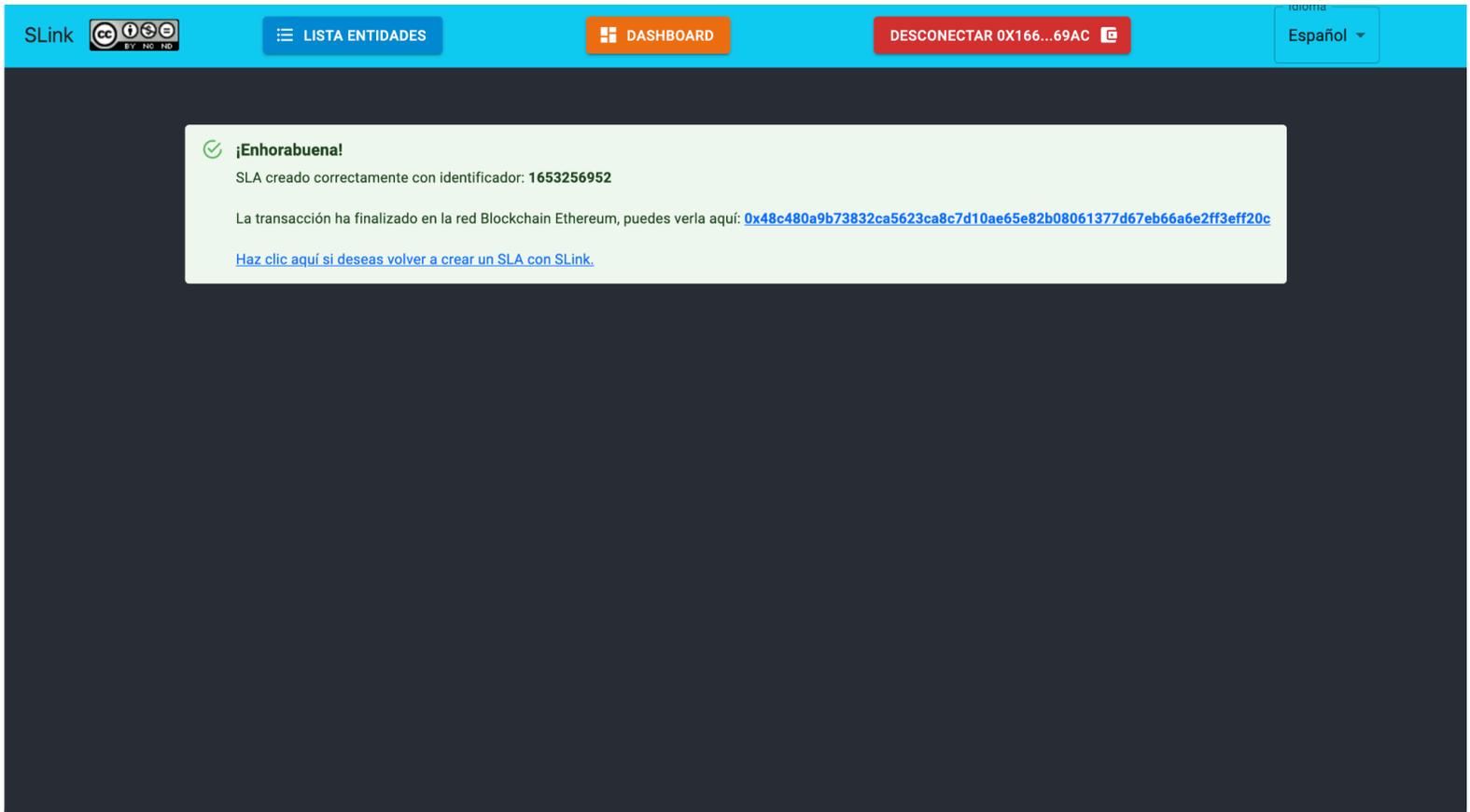


Figura 4.13 Confirmación de pago de tasas de transacción Ethereum - paso 3

4.3.5. Lista de entidades

El sistema proporciona una pantalla de gestión de entidades, siendo estas los clientes, empresas, SLAs y peticiones de contacto. Para acceder a ella, se debe pulsar el botón de la cabecera “Lista entidades”, tal y como se muestra en la [Figura 4.5](#). Este botón es visible únicamente por el proveedor de los servicios (propietario del contrato inteligente) en todas las pantallas salvo la principal. El sistema comprueba si la dirección Ethereum conectada es la misma que tiene el propietario del contrato inteligente para determinar si se trata del proveedor.

La pantalla permite gestiones distintas en función de la entidad:

- **Cientes:**
 - Creación (esta es la manera de proceder a crear un cliente nuevo partiendo, por ejemplo, de una petición de contacto para ello) (ver la [Figura 4.15](#))
 - Actualización (permite actualizar todos los datos de un cliente salvo su dirección Ethereum, debido a que es su identificador) (ver la [Figura 4.16](#))
 - Vista con todos sus datos
 - Listado (ver la [Figura 4.14](#))
 - Borrado (permite eliminar clientes si estos no tienen ningún SLA)
- **Empresas:**
 - Creación (ver la [Figura 4.18](#))
 - Actualización (permite actualizar todos los datos de una empresa salvo su CIF, debido a que es su identificador) (ver la [Figura 4.19](#))
 - Vista con todos sus datos
 - Listado (ver la [Figura 4.17](#))
 - Borrado (permite eliminar empresas si estas no tienen ningún cliente ni SLA)
- **SLAs:**
 - Vista con todos sus datos (ver la [Figura 4.21](#))
 - Listado (ver la [Figura 4.20](#))
- **Peticiones de contacto:**
 - Listado con todos los datos de cada una (ver la [Figura 4.22](#))
 - Borrado (permite eliminar una petición de contacto por diversas razones, como haber dado de alta a un cliente nuevo)

Para la creación de clientes y empresas, existen dos botones para ello en el pie de página.

SLink  Profesionales De Soluciones Tecnológicas Para SLAs LISTA ENTIDADES DASHBOARD DESCONECTAR 0X166...69AC Idioma Español

CLIENTES EMPRESAS SLAS CONTACTOS

Dirección ethereum	DNI	Apellidos	Email	Teléfono
0x039268c8837f62b7b09de4f35f770fa7a97d6b4d	54-5761340	Balfiye	dbalfiyef@gravatar.com	1234146051
0x1663b5f5A048B8bc99449Ff3E2A6E0f8e8069ac	27699988F	Velasco Santos	jovelasc@ucm.es	666555444
0x1776218e527066e46a94ad96abcb125a16e82297	68-2677849	Bracco	mbracco9@wordpress.com	8706497933
0x1f2b0755e6c9ff894432d395a8324c9226b43954	42-7100446	Coverdale	ncoverdaleb@google.com.br	8443068192
0x2496122aa269449d8c194843fb5f015ed767c8de	17-1118841	Dewerson	sdewersono@goo.ne.jp	6352653410
0x73e84531bF7B80A92812a56f6FbdC43510AFD1dD	23445454J	Robles	margarob@gmail.com	111222333
0x902e85ada0554d273d36948891f82b26fcd9eaf4	82-8124808	Pendre	apendre5@alibaba.com	2458425783
0xaf7a3bbc68d9140200457050cc1cade4bac87b34	32-6129380	Mullany	pmullanym@wunderground.com	8083571989
0xc81f04fe239e16d0b2b4ac643c25f99296eb48a	38-9689855	Innot	zinnokt@last.fm	5718071265
0xd803db876250d198a565a6fbc03c1e4ab325908	23-6582914	Sheather	esheathere@umich.edu	2529843001
0xe72c2d8b276b9aea7b5ba4e70cf03a998bd581b5	30-8284851	Thames	wthamesj@youtube.com	6014952836
0xee4e5966bd92561677ec1e22053a26be0e6d01f9	80-5995518	Aldred	maldred8@nba.com	7749904855
0xef2367183645bcff8470fa13e943321991ac45ec	68-1305730	Hebbard	lhebbard1@xing.com	3613100988
0xf113b49f5dcC1edF65F4392425A3aCE9c73bB8db	96040495Q	Bakes	mankes@gmail.com	999888777

+ CREAR NUEVO CLIENTE  + CREAR NUEVA EMPRESA 

Figura 4.14 Listado de clientes

SLink Profesionales De Soluciones Tecnológicas Para SLAs
 LISTA ENTIDADES
DASHBOARD
DESCONECTAR 0X166...69AC
Idioma: Español

CLIENTES EMPRESAS SLAS CONTACTOS

Dirección ethereum	DNI	Apellidos	Email	Teléfono
0x039268c8837f62b7b09de4f35f770fa7a97d6b4d	54-5761340	Balfyfe	dbalfyef@gravatar.com	1234146051
0x1663b5f5A048B8bc99449F3E2A6E0f8e8069ac	27699988F	Velasco Santos	jovelasc@ucm.es	666555444
0x1776218e527066e46a94ad96abcb125a16e82297	68-2677849	Bracco	mbracco9@wordpress.com	8706497933
0xd803db876250d198a565a6fbc03c1e4ab325908	23-6582914	Sheather	esheathere@umich.edu	2529843001
0xe72c2d8b276b9aea7b5ba4e70cf03a998bd581b5	30-8284851	Thames	wthamesj@youtube.com	6014952836
0xee4e5966bd92561677ec1e22053a26be0e6d01f9	80-5995518	Aldred	maldred8@nba.com	7749904855
0xef2367183645bcff8470fa13e943321991ac45ec	68-1305730	Hebbard	lhebbard1@xing.com	3613100988
0xf113b49f5dcC1edF65F4392425A3aCE9c73bB8db	96040495Q	Bakes	mankes@gmail.com	999888777

Crear nuevo cliente

REINICIAR
CANCELAR
GUARDAR

+ CREAR NUEVO CLIENTE
+ CREAR NUEVA EMPRESA

Figura 4.15 Creación de cliente

SLink Profesionales De Soluciones Tecnológicas Para SLAs LISTA ENTIDADES DASHBOARD DESCONECTAR 0X166...69AC Idioma Español

CLIENTES EMPRESAS SLAS CONTACTOS

Dirección ethereum	DNI	Apellidos	Email	Teléfono
0x039268c8837f62b7b09de4f35f770fa7a97d6b4d	54-5761340	Balfyfe	dbalfyef@gravatar.com	1234146051
0x1663b5f5A048B8bc99449Ff3E2A6E0f8e8069ac	27699988F	Velasco Santos	jovelasc@ucm.es	666555444
0x1776218e527066e46a94ad96abcb125a16e82297	68-2677849	Bracco	mbracco9@wordpress.com	8706497933
0x803db876250d198a565a6fbc03c1e4ab325908	23-6582914	Sheather	esheathere@umich.edu	2529843001
0xe72c2d8b276b9aea7b5ba4e70cf03a998bd581b5	30-8284851	Thames	wthamesj@youtube.com	6014952836
0xee4e5966bd92561677ec1e22053a26be0e6d01f9	80-5995518	Aldred	maldred8@nba.com	7749904855
0xef2367183645bcff8470fa13e943321991ac45ec	68-1305730	Hebbard	lhebbard1@xing.com	3613100988
0xf113b49f5dcC1edF65F4392425A3aCE9c73bB8db	96040495Q	Bakes	mankes@gmail.com	999888777

Actualizar cliente - 0x039268c8837f62b7b09de4f35f770fa7a97d6b4d

Dirección ethereum: 0x039268c8837f62b7b09de4f35f770fa7a97d6b4d

DNI: 54-5761340

Nombre: Doralyn

Apellidos: Balfyfe

Seleccionar género: Masculino

Email: dbalfyef@gravatar.com

Teléfono: 1234146051

Empresa: 203525113 (Jenkins LLC)

Provincia: 76891 Vidon Way

Ciudad: Florești

País: Tuvalu

CANCELAR GUARDAR

+ CREAR NUEVO CLIENTE

+ CREAR NUEVA EMPRESA

Figura 4.16 Actualización de cliente

SLink  Profesionales De Soluciones Tecnológicas Para SLAs [LISTA ENTIDADES](#) [DASHBOARD](#) [DESCONECTAR 0X166...69AC](#) Idioma Español

CLIENTES **EMPRESAS** SLAS CONTACTOS

CIF	Nombre empresa	Dirección empresa	
183722448	Runte, Breitenberg and Schneider	0 Hayes Park	
203525113	Jenkins LLC	980 Birchwood Drive	
287090578	Wiegand, Kuhlman and Batz	2 Larry Street	
396142563	West and Sons	33695 Spenser Center	
508539201	Lowe, Okuneva and King	9142 North Parkway	
640990100	Buckridge and Sons	771 Lyons Place	
700844186	Graham, Pfannerstill and Hagenes	433 Thompson Court	
796684131	Kunze, Reichel and Poulos	2510 Rowland Street	
832605189	Schuppe, Grant and Rippin	39 Novick Center	
893921615	Luetgten, Kreiger and Keebler	55 Forest Run Trail	
895535238	Donnelly, Russel and Heller	1199 Kedzie Street	
810968972	Manolo Bakes S.L.	Plaza España Barcelona	
U97133888	Fdi - UCM	Ciudad Universitaria Madrid	
W3592495J	Anchor Consultorias S.L.	Plaza mayor Sevilla	

[+ CREAR NUEVO CLIENTE](#) [+ CREAR NUEVA EMPRESA](#)

Figura 4.17 Listado de empresas

SLink Profesionales De Soluciones Tecnológicas Para SLAs

LISTA ENTIDADES DASHBOARD DESCONECTAR 0X166...69AC Idioma Español

CLIENTES EMPRESAS SLAS CONTACTOS

CIF	Nombre empresa	Dirección empresa	
183722448	Runte, Breitenberg and Schneider	0 Hayes Park	
203525113	Jenkins LLC	980 Birchwood Drive	
287090578	Wiegand, Kuhlman and Batz	2 Larry Street	
396142563	West and Sons	33695 Spenser Center	
5085392			
6409901			
7008441			
7966841			
832605189	Schuppe, Grant and Rippin	39 Novick Center	
893921615	Luetngen, Kreiger and Keebler	55 Forest Run Trail	
895535238	Donnelly, Russel and Heller	1199 Kedzie Street	
B10968972	Manolo Bakes S.L.	Plaza España Barcelona	
U97133888	Fdi - UCM	Ciudad Universitaria Madrid	
W3592495J	Anchor Consultorias S.L.	Plaza mayor Sevilla	

Crear nueva empresa

CIF Nombre empresa Dirección empresa

[REINICIAR](#) [CANCELAR](#) [GUARDAR](#)

+ CREAR NUEVO CLIENTE + CREAR NUEVA EMPRESA

Figura 4.18 Creación de empresa

SLink Profesionales De Soluciones Tecnológicas Para SLAs LISTA ENTIDADES DASHBOARD DESCONECTAR 0X166...69AC Idioma Español

CLIENTES **EMPRESAS** SLAS CONTACTOS

CIF	Nombre empresa	Dirección empresa	
183722448	Runte, Breitenberg and Schneider	0 Hayes Park	
203525113	Jenkins LLC	980 Birchwood Drive	
287090578	Wiegand, Kuhlman and Batz	2 Larry Street	
396142563	West and Sons	33695 Spenser Center	
5085392			
6409901			
7008441			
7966841			
832605189	Schuppe, Grant and Rippin	39 Novick Center	
893921615	Luetgten, Kreiger and Keebler	55 Forest Run Trail	
895535238	Donnelly, Russel and Heller	1199 Kedzie Street	
B10968972	Manolo Bakes S.L.	Plaza España Barcelona	
U97133888	Fdl - UCM	Ciudad Universitaria Madrid	
W3592495J	Anchor Consultorias S.L.	Plaza mayor Sevilla	

Actualizar empresa - U97133888

CIF

Nombre empresa

Dirección empresa

CANCELAR GUARDAR

+ CREAR NUEVO CLIENTE + CREAR NUEVA EMPRESA

Figura 4.19 Actualización de empresa

SLink  Profesionales De Soluciones Tecnológicas Para SLAs

[LISTA ENTIDADES](#) [DASHBOARD](#) [DESCONECTAR 0X166...69AC](#) Idioma: Español

CLIENTES EMPRESAS **SLAS** CONTACTOS

ID	Cliente	Empresa	Precio
1	0x1663b5f5A048B8bc99449Ff3E2A6E0fbfe8069ac	U97133888	10030
2	0x73e84531bF7880A92812a56f6FbdC43510AFD1dD	W3592495J	24000
4	0xf113b49f5dcC1edF65F4392425A3aCE9c73bB8db	B10968972	16340

[+ CREAR NUEVO CLIENTE](#) [+ CREAR NUEVA EMPRESA](#)

Figura 4.20 Listado de SLAs

SLink Profesionales De Soluciones Tecnológicas Para SLAs LISTA ENTIDADES DASHBOARD DESCONECTAR 0X166...69AC Idioma Español

CLIENTES EMPRESAS **SLAS** CONTACTOS

ID	Cliente	Empresa	Precio
1	0x1663b5f5A048B8bc99449Ff3E2A6E0fbfe8069ac	U97133888	10030
2	0x73e84531bF7880A92812a56f6FbdC43510AFD1dD	W3592495J	24000
4	0xf113b49f5dc1edF65F4392425A3aCF9c73b88db	B10968972	16340
16527459			35600

SLA - 1652745911

ID: 1652745911

Renovación automática: Precio: 35600

Cobertura de servicios: Gestión de infraestructura TI

Soporte extra: Soporte presencial

Horario de servicio: Tarde (15h-20h)

Pagador de licencias: Proveedor de servicios (sin costes para el...)

Periodo de reporte: Semestre

Periodo de facturación: Año

Método de facturación: Criptomonedas

[CERRAR](#)

+ CREAR NUEVO CLIENTE + CREAR NUEVA EMPRESA

Figura 4.21 Listado de datos de un SLA

SLink  Profesionales De Soluciones Tecnológicas Para SLAs

[LISTA ENTIDADES](#) [DASHBOARD](#) [DESCONECTAR 0X166...69AC](#) Idioma: [Español](#)

CLIENTES EMPRESAS SLAS **CONTACTOS**

Nombre	Apellidos	Email	Dirección Ethereum (opcional)	Asunto	Mensaje
José	Velasco Santos	jovelasc@ucm.es		Nuevo cliente	Hola! Me gustaría ser cliente vuestro :)
Yule	Zhang	yule@ucm.es	0x4b434B0745D97414387916920B68BBEB6A2fd4d8	Me gustaría probar SLink	Hola! Deseo probar SLink con sus increíbles beneficios de Blockchain.

[+ CREAR NUEVO CLIENTE](#) [+ CREAR NUEVA EMPRESA](#)

Figura 4.22 Listado de peticiones de contacto

4.3.6. Dashboard

El sistema proporciona un dashboard para visualizar los KPIs vinculados a los SLAs que permiten favorecer la medición y revisión de estos, consiguiendo así uno de los objetivos indicados en la sección [1.2. Objetivos](#). Para acceder al mismo, se debe pulsar el botón de la cabecera “Dashboard”, como se muestra en la [Figura 4.5](#). Al igual que en el apartado anterior, este botón es visible únicamente por el proveedor de los servicios (propietario del contrato inteligente) en todas las pantallas salvo la principal y para ello, la dirección Ethereum conectada deberá ser la misma que tiene el propietario del contrato inteligente para determinar si se trata del proveedor.

Como comentaba anteriormente en la sección [2.1.6. Los KPIs dentro de ITIL](#), debido a la limitación del tiempo y alcance, y la dificultad de probar el proyecto en los entornos empresariales reales, se han de definir unos KPIs propios y prácticos. Cada KPI se puede visualizar en 4 espacios temporales:

- Últimos 7 días
- Último mes
- Último año
- Desde comienzo del año en curso hasta la fecha actual (YTD por sus siglas en inglés *Year To Date*)

Los KPIs se pueden clasificar según el tipo de vista gráfica:

- **Gráfico de área:** permiten mostrar cantidades en un espacio temporal a través de un área. Solo existe un KPI para esta vista gráfica (ver la [Figura 4.23](#)):
 - Cantidad de SLAs creados a lo largo de un espacio temporal.
- **Gráfico de barras:** permiten mostrar cantidades en un espacio temporal a través de barras. Son los siguientes:
 - Cantidad de SLAs creados con un servicio determinado a lo largo de un espacio temporal (para todos los servicios configurados) (ver la [Figura 4.25](#)).
 - Cantidad de SLAs creados con un servicio extra determinado a lo largo de un espacio temporal (para todos los servicios extras configurados).
 - Cantidad de SLAs creados con un espacio de servicio determinado a lo largo de un espacio temporal (para todos los espacios de servicios configurados).

- Cantidad de SLAs creados con una licencia determinada a lo largo de un espacio temporal (para todas las licencias configuradas).
- Cantidad de SLAs creados con una periodicidad de reportes de revisión determinada a lo largo de un espacio temporal (para todas las periodicidades de reportes de revisión configuradas).
- Cantidad de SLAs creados con una periodicidad de facturación determinada a lo largo de un espacio temporal (para todas las periodicidades de facturación configuradas).
- Cantidad de SLAs creados con un métodos de facturación determinado a lo largo de un espacio temporal (para todos los métodos de facturación configurados).
- **Gráfico circular:** permite visualizar una comparación de valores entre sí. En concreto, se utilizan para ver qué cantidad de SLAs creados en un espacio temporal incluyen los elementos de configuración de un SLA (ver la [Figura 4.24](#)). Son los siguientes:
 - Distribución de servicios que incluyen los SLAs creados en un espacio temporal.
 - Distribución de servicios extra que incluyen los SLAs creados en un espacio temporal.
 - Distribución de espacios de servicio que incluyen los SLAs creados en un espacio temporal.
 - Distribución de licencias que incluyen los SLAs creados en un espacio temporal.
 - Distribución de periodicidades de reportes de revisión que incluyen los SLAs creados en un espacio temporal.
 - Distribución de periodicidades de facturación que incluyen los SLAs creados en un espacio temporal.
 - Distribución de métodos de facturación que incluyen los SLAs creados en un espacio temporal.

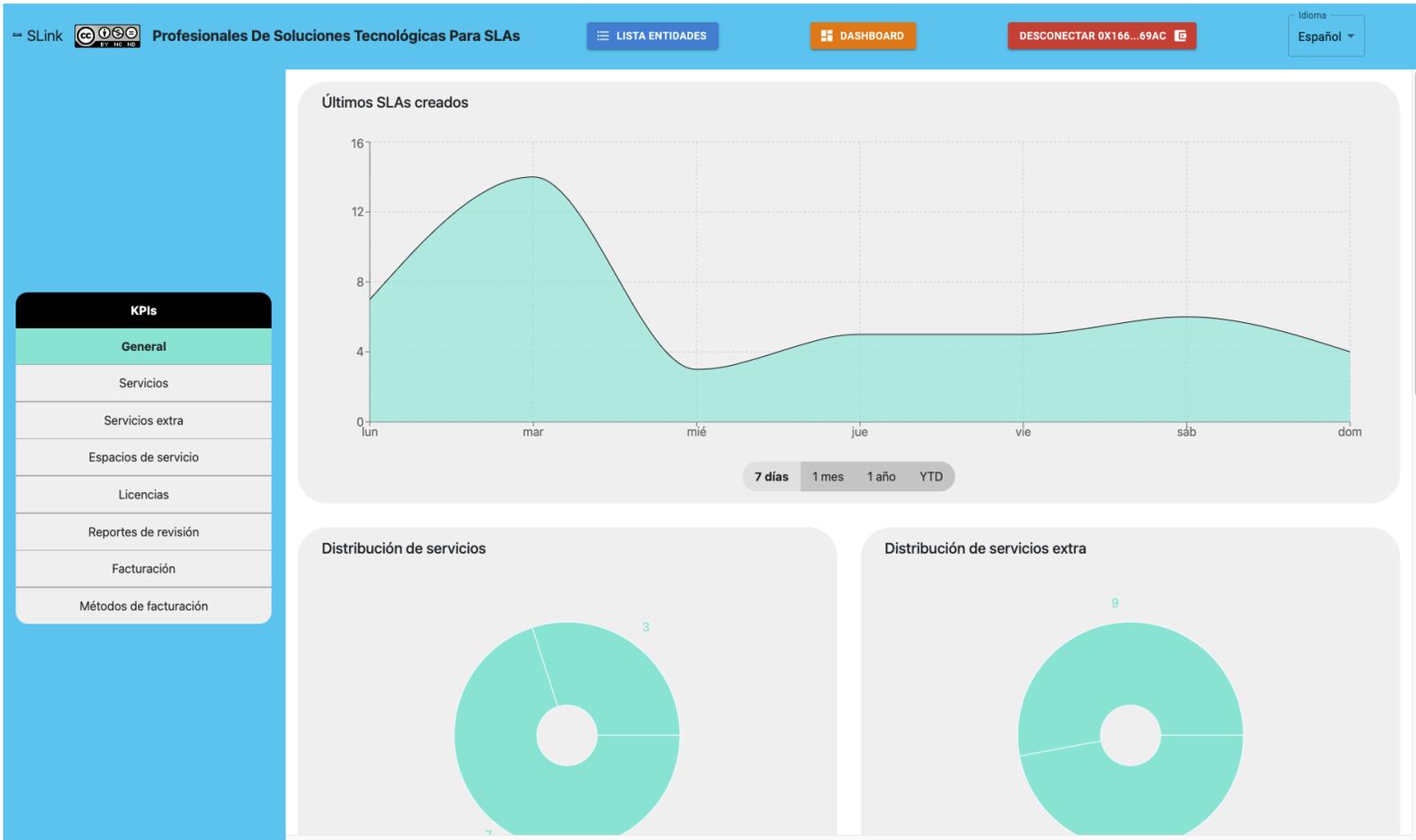


Figura 4.23 Sección general - parte 1

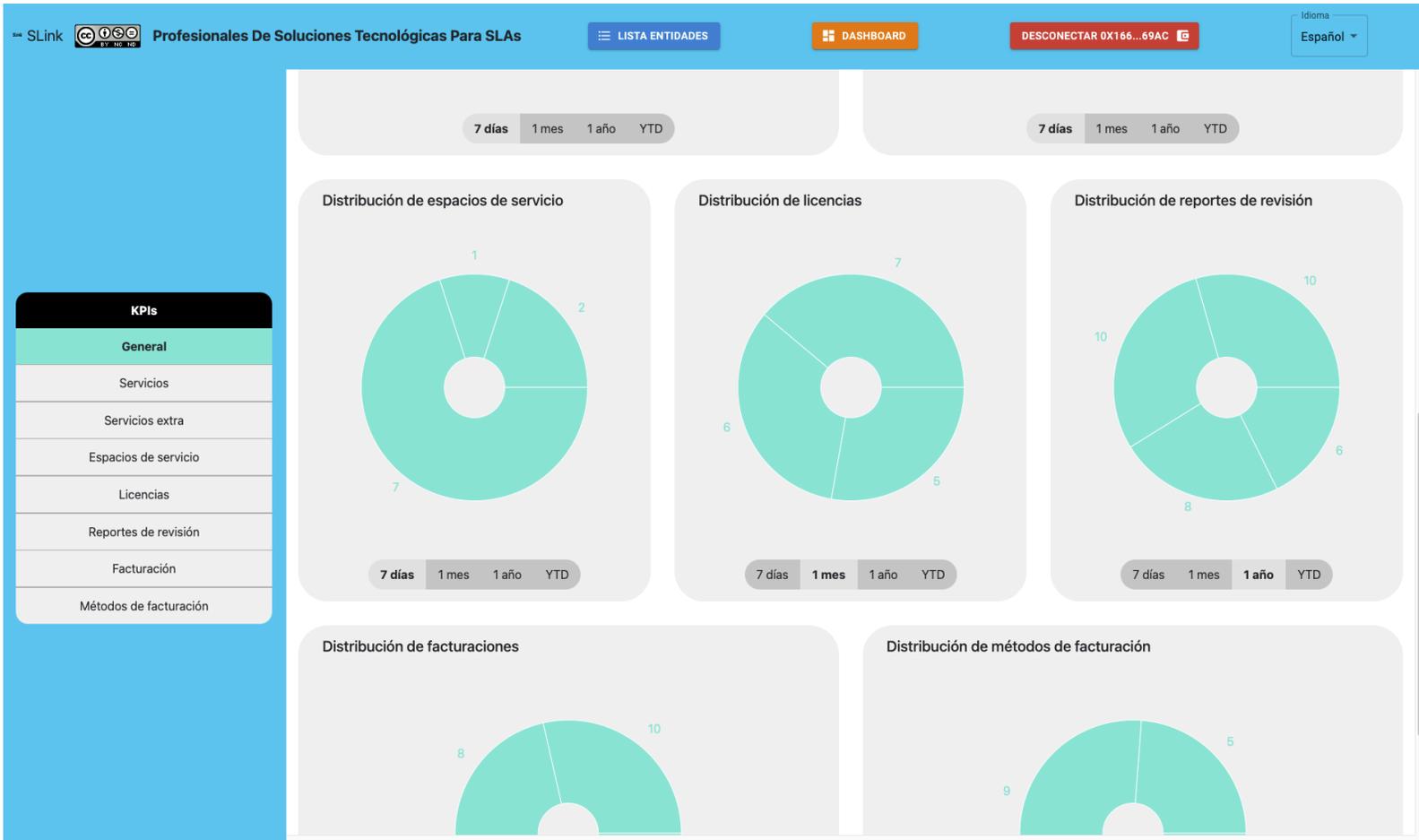


Figura 4.24 Sección general - parte 2

KPIs
General
Servicios
Servicios extra
Espacios de servicio
Licencias
Reportes de revisión
Facturación
Métodos de facturación

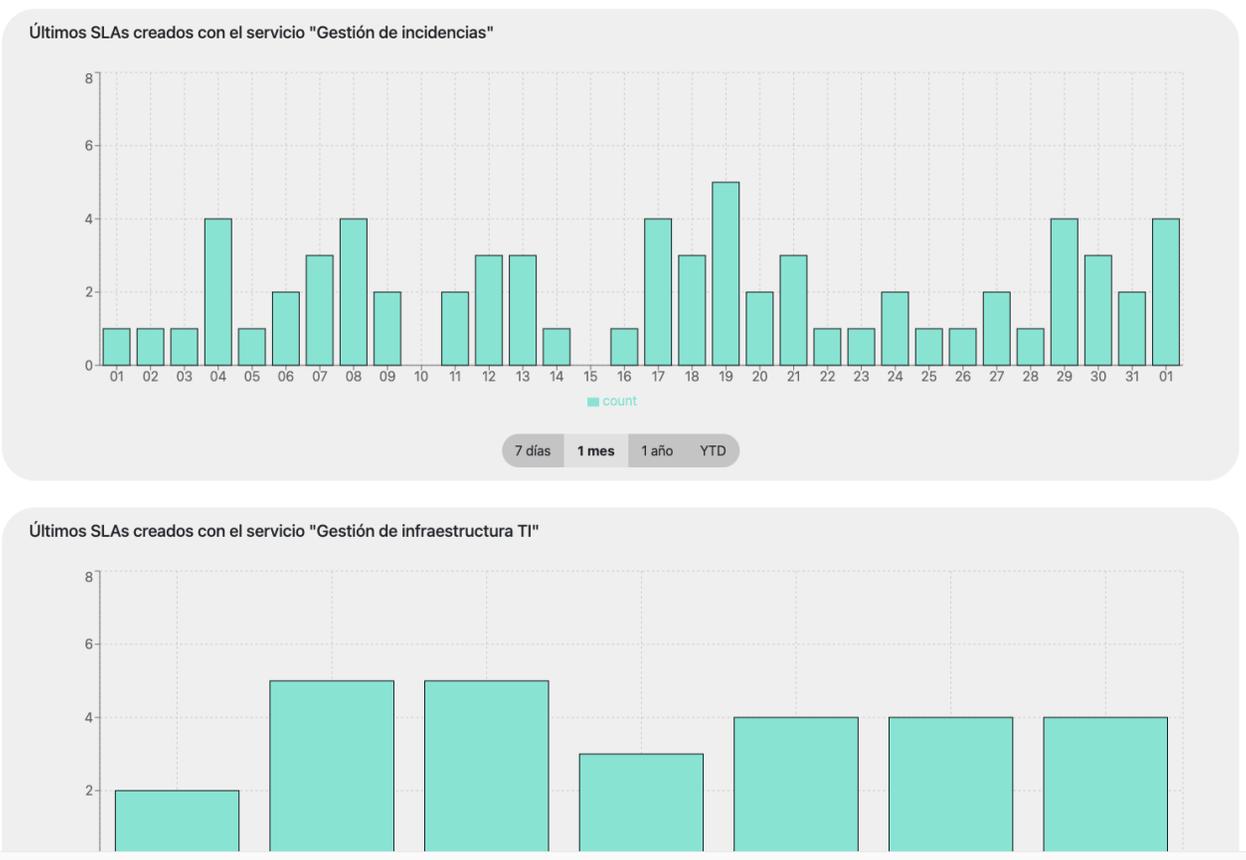


Figura 4.25 Sección servicios (un ejemplo)

4.3.7. Idiomas

El sistema SLink está disponible tanto en español como en inglés. Para cambiar de idioma, se puede hacer clic en “*Idioma*” (o “*Language*” si el sistema está en inglés) que está colocado en la parte superior-derecha de cada página (excepto la página principal, tal y como se ilustra en la [Figura 4.26](#)). En la [Figura 4.27](#) se muestra el resultado de la página después de cambiar el idioma de español al inglés, se puede observar que la página conserva el mismo aspecto que antes, y simplemente se ha traducido el texto al nuevo idioma.

 SLink. Profesionales de Soluciones Tecnológicas para SLAs

 **Oficina Central**
Av. del Partenón, 4, 28042 Madrid
Tel. 902 884 834

 **Servicio de Soporte Técnico**
Tel. 911 460 032
Email: supportTeam@slink.com

 **Centro de Operaciones**
C/ Gran Vía, 34, 28013 Madrid
Tel. 949 870 186

 **Atención Comercial**
Tel. 949 870 186 (No cliente)
Tel. 949 870 000 (Para cliente)

Contacta con nosotros

Nombre *

Apellidos *

Email *

Dirección Ethereum (opcional)

Asunto *

Mensaje *

He leído y acepto la política de privacidad
Leer el [aviso legal y la política de privacidad](#).

[Contactar](#)

Figura 4.26 Cambio de idioma

 SLink. Professionals of Technology Solutions for SLAs

 **Central Office**
Av. del Partenón, 4, 28042 Madrid
Tel. 902 884 834

 **Technical Support Service**
Tel. 911 460 032
Email: supportTeam@slink.com

 **Operation Center**
C/ Gran Vía, 34, 28013 Madrid
Tel. 949 870 186

 **Commercial Attention**
Tel. 949 870 186 (Not client)
Tel. 949 870 000 (For client)

Contact us

First name *

Last name *

Email *

Ethereum address (optional)

Subject *

Message *

I have read and accepted the policy of privacy
Read the [legal notice and privacy policy](#).

Figura 4.27 Cambio de idioma (resultado)

4.3.8. Página de Aviso legal y política de privacidad

La página de aviso legal y política de privacidad abarca la siguiente información (ver la [Figura 4.28](#)):

Aviso legal:

- La duración del defecto de SLA en el sistema es de 1 año, el cliente tiene 3 meses desde la fecha inicial para cancelarlo gratis.
- También el cliente puede cancelar cuando quiera el servicio del próximo año hasta 2 meses antes de la próxima fecha de renovación.
- La renovación automática depende totalmente del cliente.
- Todos los precios que se muestran en la página web son precios generales, si el cliente tiene una solicitud específica, puede contactar con nosotros.
- Los horarios de servicio que se muestran en la página web son horarios generales, se pueden cambiar para que se ajusten a las necesidades del cliente.

Política de privacidad:

- Nunca se compartirán los datos confidenciales del cliente con nadie más.
- El contenido de los emails de consulta y cualquier información mencionada en los mismos será leído y analizado únicamente por personales autorizados por SLink.
- Los datos sensibles incluyen: DNI, nombre y apellidos, dirección de correo electrónico, número de teléfono, datos de la empresa, dir. de Ethereum, precio total de SLA.
- Todos los datos sensibles se guardarán en una base de datos privada y no se lanzará en la Blockchain.

Se dispone de 2 maneras en el sistema para acceder a la página de Aviso legal y política de privacidad:

1. Al enviar un contacto con SLink, se obliga a aceptar la política de privacidad, se muestra (ver la [Figura 4.29](#)) un enlace para ir a esta página.
2. Al enviar un SLA a la red Ethereum a través del sistema, se necesita que el cliente acepte la política de privacidad, se muestra (ver la [Figura 4.30](#)) un enlace para ir a esta página.

Aviso legal y política de privacidad

Aviso legal

La duración del defecto de nuestro SLA es de 1 año, tiene 3 meses desde la fecha inicial para cancelarlo gratis. También puede cancelar cuando quiera el servicio del próximo año hasta 2 meses antes de la próxima fecha de renovación.

La renovación automática depende totalmente de usted.

Todos los precios que se muestran en nuestro sitio web son precios generales, si tiene una solicitud específica, no dude en contactarnos. Además, los horarios de servicio son horarios generales, podemos cambiarlos para que se ajusten a sus necesidades.

Política de privacidad

Nunca compartiremos sus datos confidenciales con nadie más. El contenido de los emails de consulta y cualquier información mencionada en los mismos será leído y analizado únicamente por personal autorizado por SLink.

Información delicada Incluye: DNI, nombre y apellidos, dirección de correo electrónico, número de teléfono, datos de la empresa, dir. de Ethereum, precio total de SLA.

Toda la información delicada se guardará en una base de datos privada y no se lanzará en la Blockchain.

Figura 4.28 Aviso legal y política de privacidad

Contacta con nosotros

Nombre *

Apellidos *

Email *

Dirección Ethereum (opcional)

Asunto *

Mensaje *

He leído y acepto la política de privacidad

Leer el [aviso legal y la política de privacidad](#).

Contactar

Figura 4.29 Acceder a Aviso legal y política de privacidad (forma 1)

Facturación de servicios

Periodo de facturación
Seleccionar periodo ...

He leído y acepto la política de privacidad
Leer el [aviso legal y la política de privacidad](#).

El precio actual de tu SLA es: 0€/year

Los campos con * son obligatorios en este formulario. El precio mostrado anteriormente no es cantidad de solicitud.

Enviar

Figura 4.30 Acceder a Aviso legal y política de privacidad (forma 2)

4.3.9. Página de Concernos

La página Concernos (ver la [Figura 4.31](#)) hace de puente entre el cliente y los personales de SLink, ofreciendo un formulario donde el cliente puede enviar peticiones de contactos al sistema y los siguientes datos de SLink:

- Oficina central (dirección y teléfono)
- Servicio de soporte técnico (teléfono y email)
- Centro de operaciones (dirección y teléfono)
- Atención comercial (teléfono)

El formulario pide la siguiente información al cliente (ver la [Figura 4.32](#)):

- Nombre
- Apellidos
- Email
- Dirección Ethereum (opcional)
- Asunto
- Mensaje

 SLink. Profesionales de Soluciones Tecnológicas para SLAs

 **Oficina Central**

Av. del Partenón, 4, 28042 Madrid
Tel. 902 884 834

 **Servicio de Soporte Técnico**

Tel. 911 460 032
Email: supportTeam@slink.com

 **Centro de Operaciones**

C/ Gran Vía, 34, 28013 Madrid
Tel. 949 870 186

 **Atención Comercial**

Tel. 949 870 186 (No cliente)
Tel. 949 870 000 (Para cliente)

Contacta con nosotros

Nombre *

Apellidos *

Email *

Dirección Ethereum (opcional)

Asunto *

Mensaje *

He leído y acepto la política de privacidad
Leer el [aviso legal y la política de privacidad](#).

Contactar

Figura 4.31 Página Conocernos

Contacta con nosotros

Nombre *

Apellidos *

Email *

Dirección Ethereum (opcional)

Asunto *

Mensaje *

He leído y acepto la política de privacidad

Leer el [aviso legal y la política de privacidad](#).

Figura 4.32 Formulario de contacto

Cabe destacar que todas las peticiones de contacto enviadas a través de este formulario aparecerán en el apartado Peticiones de contacto en la Lista de entidades (ver la sección [4.3.5. Lista de entidades](#)).

4.4. Contrato inteligente

En esta sección se expone el contenido y diseño del contrato inteligente.

Cabe mencionar parte de los tipos de datos en Solidity utilizados en la implementación del contrato inteligente del sistema:

- Enteros sin signo: uint

- Cadenas de caracteres: string
- Direcciones Ethereum: address
- Valores binarios: bool

4.4.1. Estructuras

En la [Figura 4.33](#) se muestran las estructuras existentes del contrato inteligente, las cuales están relacionadas entre ellas directamente, por ejemplo, un SLA contiene un ID (*identificador*) de servicio y por lo tanto, deberá existir un servicio con este ID. De esta manera, no existe redundancia de datos y además se reduce el coste de la transacción que crea un SLA al tener menor número de datos que guardar.

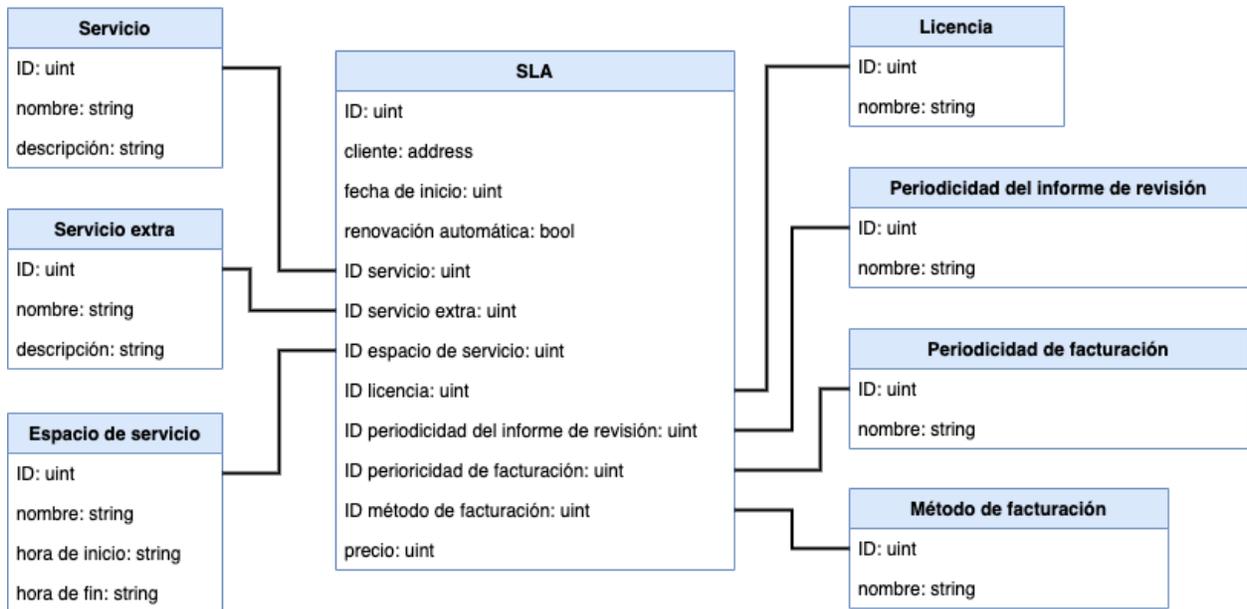


Figura 4.33 Diagrama del modelo de datos del contrato inteligente

4.4.2. Variables

Son todas aquellas variables con cualquier visibilidad que se almacenan dentro de un contrato inteligente.

Toda aquella variable pública, puede ser consultada sin la necesidad de existir una función para ello. Se muestran a continuación las variables en el contrato del sistema.

En la [Tabla 4.2](#) se muestra información sobre el atributo “Proveedor”.

Nombre	Proveedor
Tipo	Dirección Ethereum
Visibilidad	Público
Descripción	Se trata de la dirección Ethereum del creador del contrato, el considerado como propietario de este.

Tabla 4.2 Variable “Proveedor”

En la [Tabla 4.3](#) se muestra información sobre el atributo “Niveles de servicio”.

Nombre	Niveles de servicio
Tipo	Cadena de caracteres
Visibilidad	Público
Descripción	Se trata de los niveles de servicio de los SLAs que permite crear el contrato.

Tabla 4.3 Variable “Niveles de servicio”

En la [Tabla 4.4](#) se muestra información sobre el atributo “IDs de servicios”.

Nombre	IDs de servicios
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor servicio
Visibilidad	Privado
Descripción	Se trata de un diccionario donde la clave es el ID de un servicio y el valor es binario, indicando si existe o no este ID.

Tabla 4.4 Variable “IDs de servicios”

En la [Tabla 4.5](#) se muestra información sobre el atributo “Servicios”.

Nombre	Servicios
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor servicio
Visibilidad	Público
Descripción	Se trata de un diccionario donde la clave es el ID de un servicio y el valor es el servicio en concreto.

Tabla 4.5 Variable “Servicios”

En la [Tabla 4.6](#) se muestra información sobre el atributo “IDs de servicios extra”.

Nombre	IDs de servicios extra
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor binario
Visibilidad	Privado
Descripción	Se trata de un diccionario donde la clave es el ID de un servicio extra y el valor es binario, indicando si existe o no este ID.

Tabla 4.6 Variable “IDs de servicios extra”

En la [Tabla 4.7](#) se muestra información sobre el atributo “Servicios extra”.

Nombre	Servicios extra
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor servicio
Visibilidad	Público
Descripción	Se trata de un diccionario donde la clave es el ID de un servicio extra y el valor es el servicio extra en concreto.

Tabla 4.7 Variable “Servicios extra”

En la [Tabla 4.8](#) se muestra información sobre el atributo “IDs de espacios de servicios”.

Nombre	IDs de espacios de servicios
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor binario
Visibilidad	Privado
Descripción	Se trata de un diccionario donde la clave es el ID de un espacio de servicio y el valor es binario, indicando si existe o no este ID.

Tabla 4.8 Variable “IDs de espacios de servicios”

En la [Tabla 4.9](#) se muestra información sobre el atributo “Espacios de servicios”.

Nombre	Espacios de servicios
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor espacio de servicio
Visibilidad	Público

Descripción	Se trata de un diccionario donde la clave es el ID de un espacio de servicio y el valor es el espacio de servicio en concreto.
--------------------	--

Tabla 4.9 Variable “Espacios de servicios”

En la [Tabla 4.10](#) se muestra información sobre el atributo “IDs de licencias”.

Nombre	IDs de licencias
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor binario
Visibilidad	Privado
Descripción	Se trata de un diccionario donde la clave es el ID de una licencia y el valor es binario, indicando si existe o no este ID.

Tabla 4.10 Variable “IDs de licencias”

En la [Tabla 4.11](#) se muestra información sobre el atributo “Licencias”.

Nombre	Licencias
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor licencia
Visibilidad	Público
Descripción	Se trata de un diccionario donde la clave es el ID de una licencia y el valor es la licencia en concreto.

Tabla 4.11 Variable “Licencias”

En la [Tabla 4.12](#) se muestra información sobre el atributo “IDs de periodicidades de informes de revisión”.

Nombre	IDs de periodicidades de informes de revisión
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor binario
Visibilidad	Privado
Descripción	Se trata de un diccionario donde la clave es el ID de una periodicidad de informes de revisión y el valor es binario, indicando si existe o no este ID.

Tabla 4.12 Variable “IDs de periodicidades de informes de revisión”

En la [Tabla 4.13](#) se muestra información sobre el atributo “Periodicidades de informes de revisión”.

Nombre	Periodicidades de informes de revisión
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor periodicidad de informes de revisión
Visibilidad	Público
Descripción	Se trata de un diccionario donde la clave es el ID de una periodicidad de informes de revisión y el valor es la periodicidad de informes de revisión en concreto.

Tabla 4.13 Variable “Periodicidades de informes de revisión”

En la [Tabla 4.14](#) se muestra información sobre el atributo “IDs de periodicidades de facturación”.

Nombre	IDs de periodicidades de facturación
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor binario
Visibilidad	Privado
Descripción	Se trata de un diccionario donde la clave es el ID de una periodicidad de facturación y el valor es binario, indicando si existe o no este ID.

Tabla 4.14 Variable “IDs de periodicidades de facturación”

En la [Tabla 4.15](#) se muestra información sobre el atributo “Periodicidades de facturación”.

Nombre	Periodicidades de facturación
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor periodicidad de facturación
Visibilidad	Público
Descripción	Se trata de un diccionario donde la clave es el ID de una periodicidad de facturación y el valor es la periodicidad de facturación.

Tabla 4.15 Variable “Periodicidades de facturación”

En la [Tabla 4.16](#) se muestra información sobre el atributo “IDs de métodos de facturación”.

Nombre	IDs de métodos de facturación
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor binario
Visibilidad	Privado

Descripción	Se trata de un diccionario donde la clave es el ID de un método de facturación y el valor es binario, indicando si existe o no este ID.
--------------------	---

Tabla 4.16 Variable “IDs de métodos de facturación”

En la [Tabla 4.17](#) se muestra información sobre el atributo “Métodos de facturación”.

Nombre	Métodos de facturación
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor método de facturación
Visibilidad	Público
Descripción	Se trata de un diccionario donde la clave es el ID de un método de facturación y el valor es el método de facturación.

Tabla 4.17 Variable “Métodos de facturación”

En la [Tabla 4.18](#) se muestra información sobre el atributo “IDs de SLAs”.

Nombre	IDs de SLAs
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor binario
Visibilidad	Privado
Descripción	Se trata de un diccionario donde la clave es el ID de un SLA y el valor es binario, indicando si existe o no este ID.

Tabla 4.18 Variable “IDs de SLAs”

En la [Tabla 4.19](#) se muestra información sobre el atributo “SLAs”.

Nombre	SLAs
Tipo	Diccionario de pares con clave de tipo entero sin signo y valor SLA
Visibilidad	Público
Descripción	Se trata de un diccionario donde la clave es el ID de un SLA y el valor es el SLA.

Tabla 4.19 Variable “SLAs”

El motivo de utilizar diccionarios para conocer la existencia de IDs es la complejidad algorítmica. Si se tratara de un vector, la complejidad sería lineal respecto al número de IDs

existentes, mientras que con un diccionario es constante. Esto es debido a que la implementación de los diccionarios en Solidity es con tablas hash.

Solidity, a diferencia de otros lenguajes, permite consultar el valor asociado a una clave aunque no exista este par, devolviendo en tal caso el valor por defecto del tipo del valor. Por ello, si un ID no existe, devolverá falso.

4.4.3. Funciones

Las funciones del contrato inteligente se pueden resumir en que permiten añadir por un lado las opciones de elementos de los SLAs y, por otro lado, añadir los propios SLAs. Debido a que los diccionarios de elementos son públicos, no necesitan función que devuelva estos. Las funciones disponibles en el contrato inteligente son las que se muestran en la [Tabla 4.20](#).

Nombre	Descripción
Constructor	Recibe los niveles de servicio que tendrán todos los SLAs creados y obtiene la dirección que llama al constructor para almacenarla como proveedor.
Añadir servicio	<p>Recibe los datos de un servicio para añadirlo a las variables correspondientes, comprobando que no existe ya uno con el ID proporcionado.</p> <p>Emite evento informando de la creación de un servicio.</p>
Añadir servicio extra	<p>Recibe los datos de un servicio extra para añadirlo a las variables correspondientes, comprobando que no existe ya uno con el ID proporcionado.</p> <p>Emite evento informando de la creación de un servicio extra.</p>
Añadir espacio de servicio	<p>Recibe los datos de un espacio de servicio para añadirlo a las variables correspondientes, comprobando que no existe ya uno con el ID proporcionado.</p> <p>Emite evento informando de la creación de un espacio de servicio.</p>
Añadir licencia	Recibe los datos de una licencia para añadirla a las variables correspondientes, comprobando que no existe ya una con el ID proporcionado.

	Emite evento informando de la creación de una licencia.
Nombre	Descripción
Añadir periodicidad de informes de revisión	<p>Recibe los datos de una periodicidad de informes de revisión para añadirla a las variables correspondientes, comprobando que no existe ya una con el ID proporcionado.</p> <p>Emite evento informando de la creación de una periodicidad de informes de revisión.</p>
Añadir periodicidad de facturación	<p>Recibe los datos de una periodicidad de facturación para añadirla a las variables correspondientes, comprobando que no existe ya una con el ID proporcionado.</p> <p>Emite evento informando de la creación de una periodicidad de facturación.</p>
Añadir método de facturación	<p>Recibe los datos de un método de facturación para añadirlo a las variables correspondientes, comprobando que no existe ya uno con el ID proporcionado.</p> <p>Emite evento informando de la creación de un método de facturación.</p>
Añadir SLA	<p>Recibe los datos de un SLA para añadirlo a las variables correspondientes, comprobando que no existe ya uno con el ID proporcionado.</p> <p>Emite evento informando de la creación de un SLA.</p>

Tabla 4.20 Funciones del contrato inteligente

4.4.4. Eventos

Los eventos son la forma que tiene un contrato inteligente de informar de hechos ocurridos dentro de sí mismo. La forma de comunicarse desde fuera de un contrato inteligente hacia estos, son las funciones disponibles en los propios contratos inteligentes y, los eventos, son la forma de comunicarse en dirección contraria. De esta manera, si el contrato inteligente lo permite, se puede establecer una comunicación bidireccional desde fuera con estos.

En el caso del sistema, los eventos pueden ser utilizados, por ejemplo, cuando un nuevo Servicio es añadido al contrato inteligente. Esto permitirá informar de la existencia de un nuevo Servicio disponible para la creación de SLAs.

En general, el contrato inteligente del sistema informa de lo que ocurre en todas las funciones salvo en el constructor.

Los eventos se muestran en la siguiente [Tabla 4.21](#):

Nombre	Parámetros
ServiceCreated	ID (uint), nombre (string) y descripción (string) del servicio.
ExtraServiceCreated	ID (uint), nombre (string) y descripción (string) del servicio extra.
ServiceSpaceCreated	ID (uint), nombre (string), inicio (string) y fin (string) del espacio de servicio.
LicenseCreated	ID (uint) y nombre (string) de la licencia.
RevisionReportCreated	ID (uint) y nombre (string) de la periodicidad de informes de revisión.
BillingCreated	ID (uint) y nombre (string) de la periodicidad de facturación.
BillingMethodCreated	ID (uint) y nombre (string) del método de facturación.
SLACreated	ID (uint), cliente (address), fecha de inicio (uint), elección de renovación automática (bool), ID del servicio (uint), ID del servicio extra (uint), ID del espacio de servicio (uint), ID de la licencia (uint), ID de la periodicidad de informes de revisión (uint), ID de la periodicidad de facturación (uint), ID del método de facturación (uint) y el precio del SLA (uint).

Tabla 4.21 Eventos del contrato inteligente

4.5. Back-end

En esta sección se expondrán todas las tecnologías y funcionalidades que se pueden observar en el back-end del sistema SLink.

Algunas tareas del back-end son imprescindibles, como todas las que tienen que ver con operaciones que se realizan sobre la base de datos privada. Otras tareas, como por ejemplo,

las que están relacionadas con la obtención de datos desde The Graph, son opcionales por que se podrían realizar desde el front-end.

Aprovechando la existencia de un back-end para las tareas imprescindibles (todas aquellas tareas que no son llamadas a la API de The Graph), se ha decidido utilizarlo también para las tareas opcionales (llamadas a la API de The Graph), ya que al delegar casi todas las operaciones al back-end se simplifica la arquitectura y se centraliza el acceso a todos los datos del sistema. Aun así, sigue existiendo una operación que se realiza sin tener al back-end de intermediario: la creación de un SLA en la blockchain.

4.5.1. Base de datos privada

Existen datos considerados sensibles en un SLA, siendo estos:

- **Datos relativos al cliente:**
 - DNI
 - Nombre
 - Apellidos
 - Email
 - Teléfono
 - Género
 - Provincia
 - Ciudad
 - País
- **Datos relativos a la empresa:**
 - CIF
 - Nombre
 - Dirección postal
- **Datos relativos al SLA:**
 - Empresa del cliente en el momento de la creación de un SLA.
 - Precios de servicios, servicios extra, espacios de servicios, licencias
 - Periodicidad de informes de revisión, periodicidades de facturación
 - Métodos de facturación

Todos ellos se guardan en la base de datos privada, salvo el precio de las opciones de un SLA que estará en el fichero de configuración del sistema (src/config.js en el repositorio del código fuente del front-end).

Un cliente puede cambiar de empresa, pero no se puede perder la vinculación a las empresas que pudiera tener anteriormente si creó algún SLA con ellas. Por ello, la base de datos privada almacena la vinculación de un SLA con su cliente, empresa y precio total. Este último campo es opcional, pero por cuestiones de rendimiento se almacena dentro de la base de datos privada, ya que estos datos son los mostrados para los SLAs creados en el listado de entidades.

Además, debido a que la forma de contacto pasa por rellenar el formulario para tal finalidad, la base de datos privada contiene la información acerca de un contacto:

- Nombre
- Apellidos
- Email
- Dirección de Ethereum (opcional)
- Motivo de contacto
- Mensaje

Se puede conocer más en detalle en la [Figura 4.34](#).

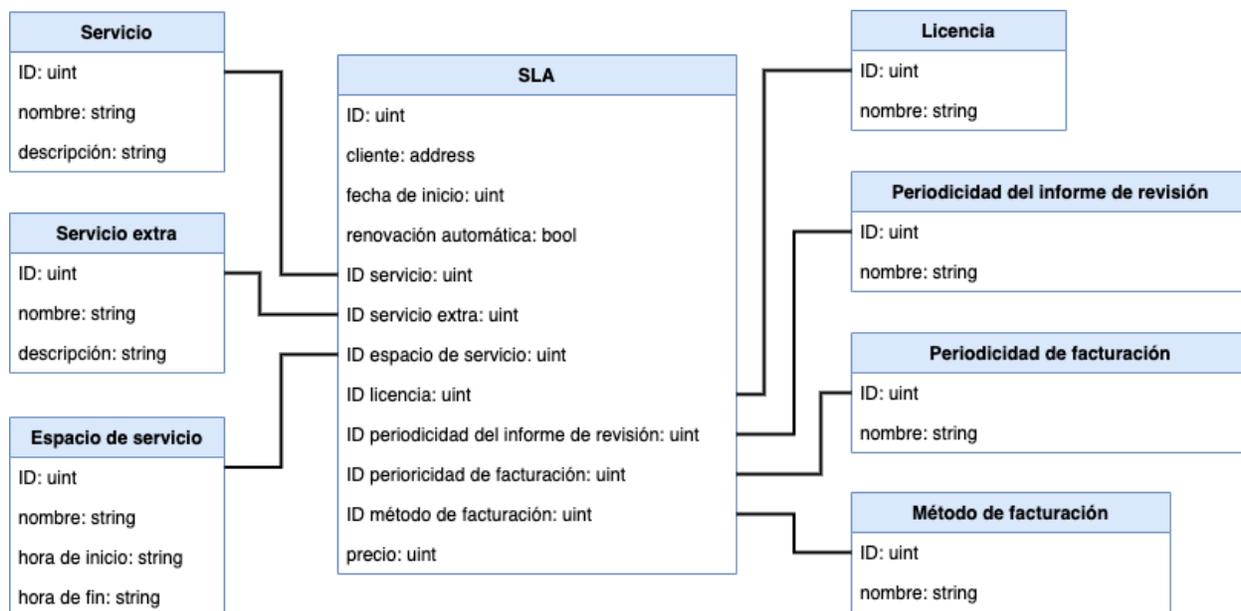


Figura 4.35 Diagrama de clases de The Graph

4.5.3. API REST

Con todo lo mencionado en los puntos anteriores, el back-end proporciona una Interfaz de Programación de Aplicaciones (del inglés *Application Programming Interfaces*, *API*) que proporciona un mecanismo de comunicación entre front-end y el propio back-end.

Esta API es de tipo Transferencia de Estado Representacional (REST por sus siglas en inglés, *Representational State Transfer*). REST es un estilo de arquitectura software para el desarrollo web que sigue unos principios como:

- Arquitectura cliente-servidor (separación de responsabilidades y portabilidad).
- Ausencia de estado (el estado lo posee el cliente y no el servidor, de manera que este debe proporcionar toda la información necesaria al servidor en una solicitud).
- Sistemas por capas (el cliente debe tener únicamente el conocimiento de la capa a la que está hablando).

Una petición a la API REST del sistema debe ajustarse a un verbo (utilizado para diferenciar acciones con una misma dirección), una URL y quizás a unos parámetros de consulta al final de la URL. Existen gran cantidad de verbos pero los utilizados en el sistema son:

- GET (consulta de recursos)

- POST (creación de recursos)
- PUT (actualización de recursos)
- DELETE (eliminación de recursos)

En la [Tabla 4.22](#) se muestra la API REST que ofrece el sistema para la gestión de clientes.

Verbo	POST
URL	customers?ethAddress=A&name=B&surname=C&dni=D&email=E&phone=F&province=G&city=H&company=I
Descripción	Creación de un cliente donde: <ul style="list-style-type: none"> • A es la dirección de Ethereum • B es el nombre • C es el apellido • D es el DNI • E es el email • F es el número de teléfono • G es la provincia • H es la ciudad • I es la compañía
Verbo	GET
URL	customer?A
Descripción	Consulta de un cliente donde A es la dirección de Ethereum.
Verbo	GET
URL	customers
Descripción	Consulta de todos los clientes.
Verbo	PUT
URL	customers/A?&name=B&surname=C&dni=D&email=E&phone=F&province=G&city=H&company=I
Descripción	Modificación de un cliente donde: <ul style="list-style-type: none"> • A es la dirección de Ethereum • B es el nombre • C es el apellido • D es el DNI • E es el email • F es el número de teléfono • G es la provincia • H es la ciudad

	<ul style="list-style-type: none"> • I es la compañía
Verbo	DELETE
URL	customers/A
Descripción	Borrado de un cliente donde A es la dirección de Ethereum.

Tabla 4.22 API REST para gestión de clientes

En la [Tabla 4.23](#) se muestra la API REST que ofrece el sistema para la gestión de empresas.

Verbo	POST
URL	companies?cif=A&name=B&address=C
Descripción	Creación de una empresa donde: <ul style="list-style-type: none"> • A es el CIF • B es el nombre • C es la dirección postal
Verbo	GET
URL	companies?A
Descripción	Consulta de una empresa donde A es el CIF.
Verbo	GET
URL	companies
Descripción	Consulta de todas las empresas.
Verbo	PUT
URL	companies?cif=A&name=B&address=C
Descripción	Modificación de una empresa donde: <ul style="list-style-type: none"> • A es el CIF • B es el nombre • C es la dirección postal
Verbo	DELETE
URL	companies/A
Descripción	Borrado de una empresa donde A es el CIF.

Tabla 4.23 API REST para gestión de empresas

En la [Tabla 4.24](#) se muestra la API REST que ofrece el sistema para la gestión de SLAs.

Verbo	POST
URL	slas?id=A&customer=B&company=C
Descripción	<p>Creación de un SLA donde:</p> <ul style="list-style-type: none"> • A es el ID • B es la dirección Ethereum del cliente • C es el CIF de la empresa <p>Recibe únicamente los datos sensibles, el resto se encuentra en Blockchain.</p>
Verbo	GET
URL	slas?A
Descripción	<p>Consulta de un SLA donde A es el ID.</p> <p>Devuelve todos los datos sensibles de un SLA.</p>
Verbo	GET
URL	theGraph/slas?A
Descripción	<p>Consulta de un SLA donde A es el ID.</p> <p>Devuelve todos los datos públicos almacenados en la red de Ethereum de un SLA.</p>
Verbo	GET
URL	slas
Descripción	Devuelve todos los datos sensibles de todos los SLAs.
Verbo	GET
URL	theGraph/slas?start=A&end=B
Descripción	<p>Consulta de todos los SLAs que estén entre A y B (fechas y horas en milisegundos opcionales).</p> <p>Devuelve todos los datos públicos almacenados en la red de Ethereum de los SLAs.</p>

Tabla 4.24 API REST para gestión de SLAs

En la [Tabla 4.25](#) se muestra la API REST que ofrece el sistema para la gestión de peticiones de contacto.

Verbo	POST
URL	contactRequests?name=A&surname=B&email=C&ethAddress=D&subject=E&message=F
Descripción	Creación de una petición de contacto donde: <ul style="list-style-type: none"> • A es el nombre • B es el apellido • C es el email • D es la dirección de Ethereum • E es el motivo de contacto • F es el mensaje
Verbo	GET
URL	contactRequests
Descripción	Consulta de todas las peticiones de contacto
Verbo	DELETE
URL	contactRequests?A
Descripción	Borrado de una petición de contacto donde A es el ID.

Tabla 4.25 API REST para gestión de peticiones de contacto

En la [Tabla 4.26](#) se muestra la API REST que ofrece el sistema para la gestión de servicios de SLA.

Verbo	GET
URL	theGraph//services/A?start=B&end=C
Descripción	Consulta un servicio, de los SLAs que lo incluyan y que estén entre B y C (fechas y horas en milisegundos opcionales).
Verbo	GET
URL	theGraph//services?start=A&end=B
Descripción	Consulta de todos los servicios, de los SLAs que los incluyan y que estén entre A y B (fechas y horas en milisegundos opcionales).

Tabla 4.26 API REST para gestión de servicios de SLA

En la siguiente [Tabla 4.27](#) se muestra la API REST que ofrece el sistema para la gestión de servicios extra de SLA.

Verbo	GET
URL	theGraph//extraServices/A?start=B&end=C
Descripción	Consulta un servicio extra, de los SLAs que lo incluyan y que estén entre B y C (fechas y horas en milisegundos opcionales).
Verbo	GET
URL	theGraph//extraServices?start=A&end=B
Descripción	Consulta de todos los servicios extra, de los SLAs que los incluyan y que estén entre A y B (fechas y horas en milisegundos opcionales).

Tabla 4.27 API REST para gestión de servicios extra de SLA

En la [Tabla 4.28](#) se muestra la API REST que ofrece el sistema para la gestión de espacios de servicios de SLA.

Verbo	GET
URL	theGraph//serviceSpaces/A?start=B&end=C
Descripción	Consulta un espacio de servicio, de los SLAs que lo incluyan y que estén entre B y C (fechas y horas en milisegundos opcionales).
Verbo	GET
URL	theGraph//serviceSpaces?start=A&end=B
Descripción	Consulta de todos los espacios de servicio, de los SLAs que los incluyan y que estén entre A y B (fechas y horas en milisegundos opcionales).

Tabla 4.28 API REST para gestión de espacios de servicios de SLA

En la [Tabla 4.29](#) se muestra la API REST que ofrece el sistema para la gestión de licencias de SLA.

Verbo	GET
URL	theGraph//licenses/A?start=B&end=C
Descripción	Consulta una licencia, de los SLAs que la incluyan y que estén entre B y C (fechas y horas en milisegundos opcionales).

Verbo	GET
URL	theGraph//licenses?start=A&end=B
Descripción	Consulta de todas las licencias, de los SLAs que las incluyan y que estén entre A y B (fechas y horas en milisegundos opcionales).

Tabla 4.29 API REST para gestión de licencias de SLA

En la [Tabla 4.30](#) y la [Tabla 4.31](#) se muestra la API REST que ofrece el sistema para la gestión de reportes de revisión y facturación de SLA.

Verbo	GET
URL	theGraph//revisionReport/A?start=B&end=C
Descripción	Consulta una periodicidad de informes de revisión, de los SLAs que la incluyan y que estén entre B y C (fechas y horas en milisegundos opcionales).
Verbo	GET
URL	theGraph//revisionReport?start=A&end=B
Descripción	Consulta de todas las periodicidades de informes de revisión, de los SLAs que las incluyan y que estén entre A y B (fechas y horas en milisegundos opcionales).

Tabla 4.30 API REST para gestión de reportes de revisión de SLA

Verbo	GET
URL	theGraph//billings/A?start=B&end=C
Descripción	Consulta una periodicidad de facturación, de los SLAs que la incluyan y que estén entre B y C (fechas y horas en milisegundos opcionales).
Verbo	GET
URL	theGraph//billings?start=A&end=B
Descripción	Consulta de todas las periodicidades de facturación, de los SLAs que los incluyan y que estén entre A y B (fechas y horas en milisegundos opcionales).

Tabla 4.31 API REST para gestión de facturación de SLA

En la [Tabla 4.32](#) se muestra la API REST que ofrece el sistema para la gestión de métodos de facturación de SLA.

Verbo	GET
URL	theGraph//billingMethods/A?start=B&end=C
Descripción	Consulta un método de facturación, de los SLAs que lo incluyan y que estén entre B y C (fechas y horas en milisegundos opcionales).
Verbo	GET
URL	theGraph//billingMethods?start=A&end=B
Descripción	Consulta de todos los métodos de facturación, de los SLAs que los incluyan y que estén entre A y B (fechas y horas en milisegundos opcionales).

Tabla 4.32 API REST para gestión de métodos de facturación de SLA

4.6. Despliegue del sistema

Los pasos necesarios para realizar el despliegue del sistema requieren de un conocimiento y dominio avanzado de prácticamente todas las tecnologías utilizadas en el proyecto.

En los siguientes apartados se muestran todos los pasos.

4.6.1. Despliegue del contrato inteligente

El primer paso es disponer de la extensión MetaMask instalada con una dirección Ethereum dentro. Esta dirección debe contener una cantidad mínima de Ether.

Para el desarrollo de este TFG, se ha utilizado una red de pruebas de Ethereum llamada Rinkeby²⁷. De cara al pago de transacciones, existen diversas formas de obtener Ether en la red de pruebas Rinkeby, una de ellas es a través del siguiente enlace: <https://faucets.chain.link/rinkeby>.

El despliegue del contrato inteligente se realiza en el entorno de desarrollo integrado Remix²⁸.

Para ello, se debe ir a su página web y ahí, conectar un monedero MetaMask con una dirección que será el creador del contrato inteligente (proveedor de servicios).

²⁷ <https://www.rinkeby.io/#stats>

²⁸ <https://remix.ethereum.org/>

Después, se ha de copiar y pegar el código del contrato inteligente del sistema para poder compilarlo. Para poder llegar a este punto, se debe hacer clic en la tercera pestaña lateral izquierda. En el caso del contrato inteligente del sistema, se recomienda compilar con la versión 0.6.12 del compilador, tal y como se muestra en la [Figura 4.36](#).

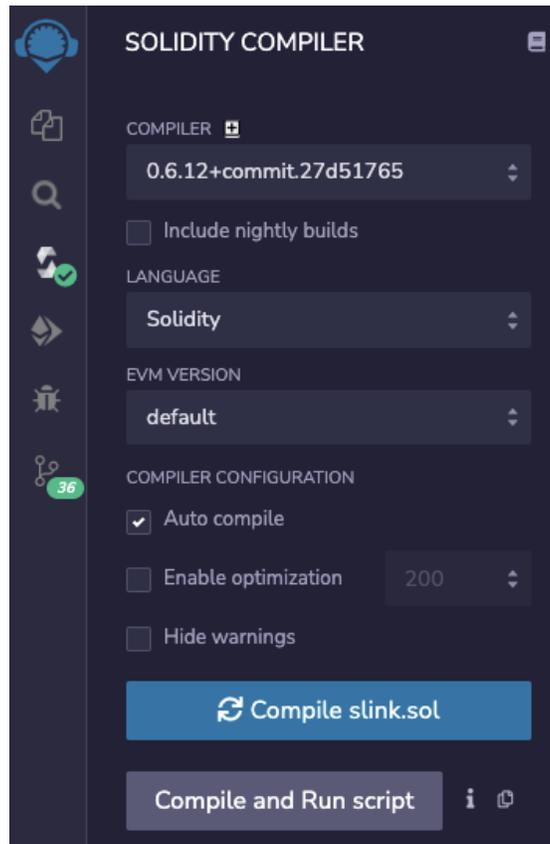


Figura 4.36 Compilar el contrato inteligente

Una vez compilado, se puede desplegar. Para ello, se debe hacer clic en la cuarta pestaña lateral izquierda. Se ha de seleccionar el entorno “*Injected Web3*” que permitirá la creación del contrato inteligente a partir de MetaMask, con la dirección que esté activa y en la red conectada como se muestra en la [Figura 4.37](#). También se debe seleccionar el contrato SLink y proporcionar los niveles de servicio correspondientes. Una vez esté todo configurado como se puede ver en la [Figura 4.38](#), haciendo clic en “*Deploy*”, se desplegará el contrato inteligente.

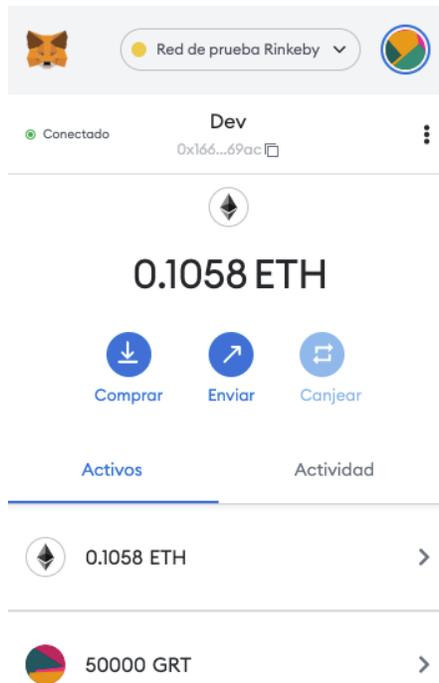


Figura 4.37 MetaMask con Rinkeby

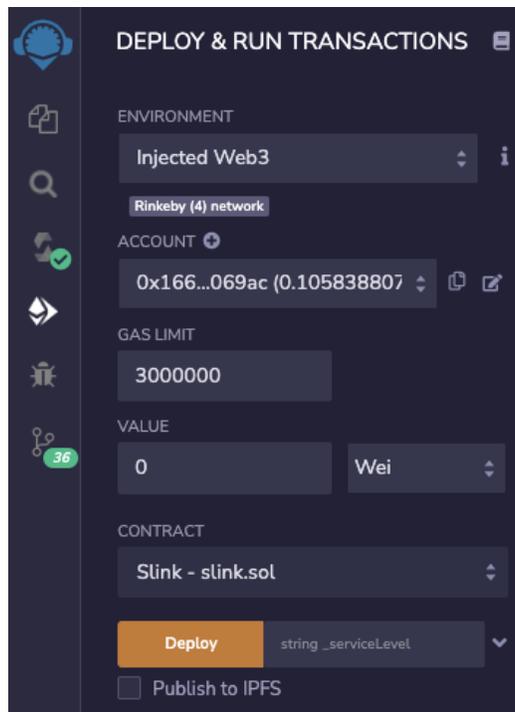


Figura 4.38 Opciones de despliegue

Una vez el contrato inteligente se ha desplegado, aparecerán justo debajo de la imagen anterior una serie de botones y campos de texto para llamar a las funciones y variables del contrato inteligente como se puede apreciar en la [Figura 4.39](#).

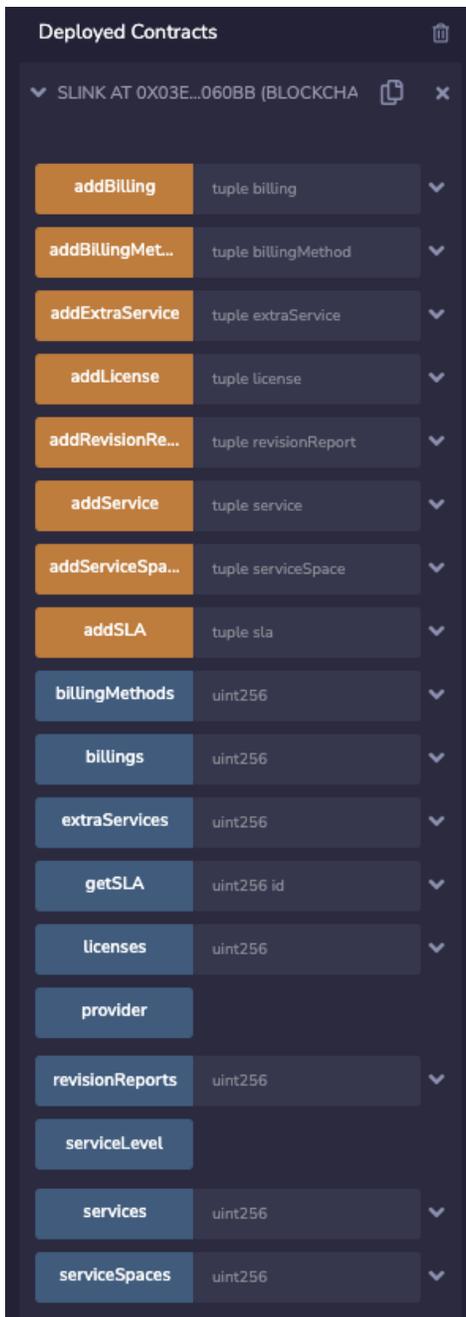


Figura 4.39 Funciones y variables del contrato inteligente desplegado

Cada una de las opciones del SLA se han de añadir con la función correspondiente. Por ejemplo, para añadir un servicio disponible en una configuración determinada, se debe introducir en el campo de texto al lado del botón “addService” el servicio para, posteriormente, hacer clic en el botón “transact” como se muestra en la [Figura 4.40](#). Esto llevará al pago de las tasas desde MetaMask y creará la transacción.

La sintaxis para introducir un valor de tipo estructura en Remix consiste en englobar todo entre corchetes y separar los distintos valores entre comas. Los valores de tipo string, se engloban entre dobles comillas.

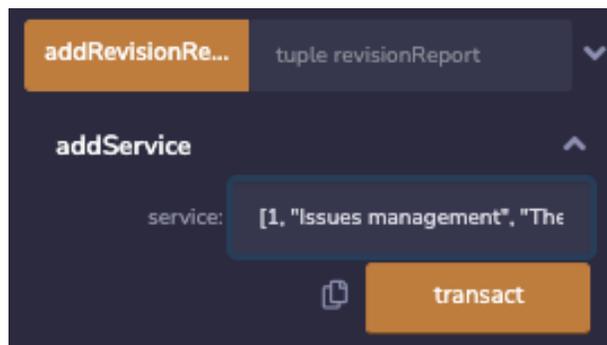


Figura 4.40 Llamada a una función del contrato inteligente desplegado

4.6.2. Despliegue del subgrafo de The Graph

Para el despliegue del subgrafo en The Graph se utiliza Subgraph Studio²⁹. Para ello, se debe conectar el monedero MetaMask (no hace falta que sea la misma dirección propietario del contrato inteligente) en su página web, hacer clic en Productos y en Subgraph Studio. Allí, con el botón “*Crear un Subgrafo*” disponible arriba a la derecha como se aprecia en la [Figura 4.41](#), tendremos la opción de crear un subgrafo para la red deseada con el nombre buscado como se puede ver en la [Figura 4.42](#).

²⁹ <https://thegraph.com/studio/>

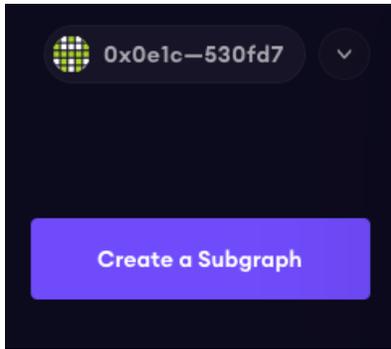


Figura 4.41 Botón de creación de un subgrafo

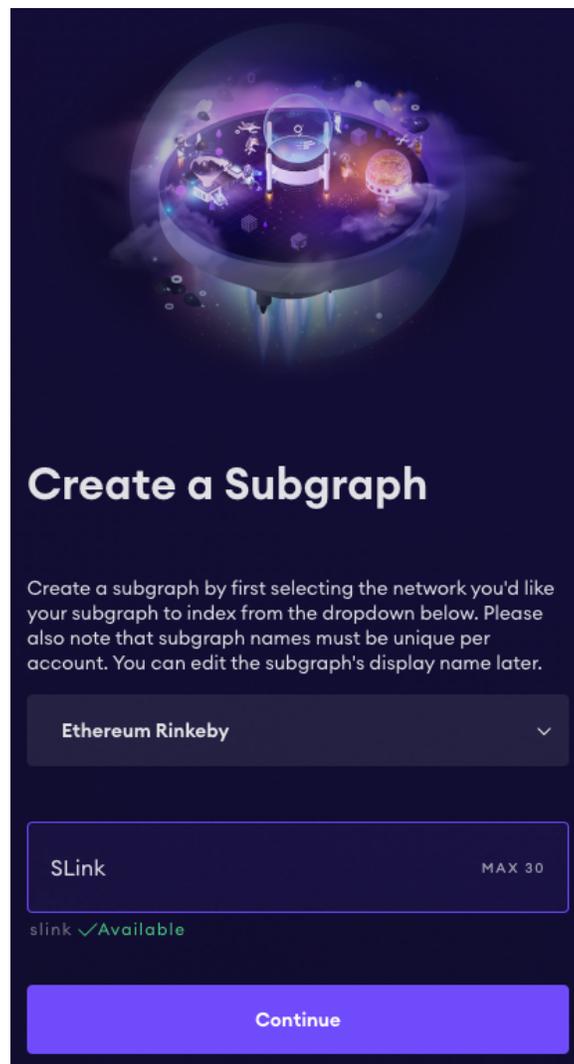


Figura 4.42 Opciones a configurar en la creación de un subgrafo

Los datos que muestra la pantalla del subgrafo (como la [Figura 4.43](#)) son importantes para los siguientes pasos:

- Subgraph SLUG: se trata del identificador del subgrafo.
- Deploy key: es la clave que permite realizar el despliegue.

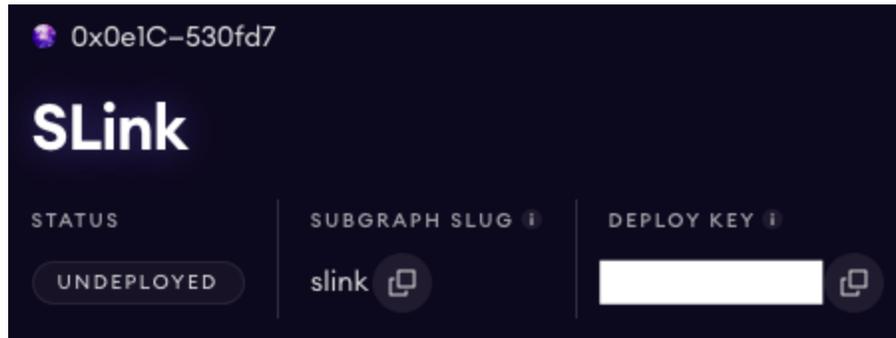


Figura 4.43 Datos del subgrafo

El siguiente paso es proporcionar el código del contrato inteligente a Etherscan³⁰ ya que en los siguientes pasos, será necesario que Etherscan conozca el código del contrato, por lo que se ha de proporcionar.

En la dirección <https://etherscan.io/verifyContract> podemos realizar este proceso. Si el despliegue es en la red de Rinkeby, el enlace sería <https://rinkeby.etherscan.io/verifyContract>.

En esta página se ha de proporcionar la dirección del contrato, el tipo de compilación (un solo archivo de Solidity), la versión del compilador (0.6.12) y el tipo de licencia (MIT³¹) como se puede apreciar en la [Figura 4.44](#).

³⁰ <https://etherscan.io/>

³¹ https://es.wikipedia.org/wiki/Licencia_MIT

Etherscan
Rinkeby Testnet Network

All Filters Search by Address / Txn Hash / Block / Token / Ens

Home Blockchain Tokens Misc Rinkeby

Verify & Publish Contract Source Code

COMPILER TYPE AND VERSION SELECTION

Source code verification provides **transparency** for users interacting with smart contracts. By uploading the source code, Etherscan will match the compiled code with that on the blockchain. Just like contracts, a "smart contract" should provide end users with more information on what they are "digitally signing" for and give users an opportunity to audit the code to independently verify that it actually does what it is supposed to do.

Please enter the Contract Address you would like to verify

0x03eeC2cc89624169951C03d2379fefBd5fc060BB

Please select Compiler Type

Solidity (Single file)

Please select Compiler Version

v0.6.12+commit.27d51765

Un-Check to show all nightly Commits also

Please select Open Source License Type

3) MIT License (MIT)

I agree to the [terms of service](#)

Continue Reset

Figura 4.44 Verificación y publicación del código de un contrato inteligente

Posteriormente, se proporciona el código del contrato inteligente y si es el correcto, el código ya estará en Etherscan, siendo este visible a cualquier persona desde su página web como se puede ver en la [Figura 4.45](#).

Contract 0x03eeC2cc89624169951C03d2379fefBd5fc060BB

Contract Overview

Balance: 0 Ether

More Info

More

My Name Tag: Not Available

Creator: 0x1663b5f5a048b8bc99... at txn 0x131486ee00c9a2a0f8...

Transactions Contract Events

Code Read Contract Write Contract

Search Source Code

Contract Source Code Verified (Exact Match)



Contract Name: Slink

Optimization Enabled: No with 200 runs

Compiler Version v0.6.12+commit.27d51765

Other Settings: default evmVersion, MIT license

Contract Source Code (Solidity)

Outline More Options



```
1 - ✓**
2  *Submitted for verification at Etherscan.io on 2022-05-16
3  */
4
5  pragma solidity >=0.4.4 < 0.7.0;
6  pragma experimental ABIEncoderV2;
7
8  // @title Slink
9  // @author Yule Zhang, José Andrés Velasco Santos
```

Figura 4.45 Código verificado y publicado de un contrato inteligente

Una vez el subgrafo está creado y el código del contrato inteligente está desplegado en Etherscan, estos son los pasos a seguir:

1. Instalación de The Graph CLI (necesario tener npm o yarn instalado, ambos son gestores de paquete Javascript) como se puede ver en la [Figura 4.46](#).

```
# NPM
$ npm install -g @graphprotocol/graph-cli

# Yarn
$ yarn global add @graphprotocol/graph-cli
```

Figura 4.46 Instalación de The Graph CLI³²

2. Inicialización del subgrafo como se puede visualizar en la [Figura 4.47](#).

```
graph init --studio <SUBGRAPH_SLUG>
```

Figura 4.47 Inicialización del subgrafo³³

3. Escritura del subgrafo: el comando anterior crea una base de estructura para el subgrafo, a partir del código público en Etherscan. En el caso de SLink, esto no es suficiente y se debe dar forma al modelo de datos como se indica en la sección [4.5.2. The Graph en el back-end](#) editando los archivos:
 - a. Manifiesto (subgraph.yaml): define qué fuentes de datos indexará el subgrafo.
 - b. Esquema (schema.graphql): define los datos a recuperar de un subgrafo.
 - c. Mapeador (mapping.ts): traduce los datos de las fuentes de datos a las entidades definidas en el esquema.

Todos estos archivos mantienen el mismo contenido siempre salvo dos cambios necesarios en el despliegue dentro del manifiesto. Estos son la dirección del contrato y el bloque de inicio (bloque de la red de Ethereum donde se creó el contrato inteligente).

4. Despliegue del subgrafo como se muestra en la [Figura 4.48](#) y [Figura 4.49](#).

³² <https://thegraph.com/docs/es/developer/quick-start/>

³³ <https://thegraph.com/docs/es/developer/quick-start/>

```
$ graph codegen
$ graph build
```

Figura 4.48 Generación y construcción del código de un subgrafo³⁴

```
$ graph auth --studio <DEPLOY_KEY>
$ graph deploy --studio <SUBGRAPH_SLUG>
```

Figura 4.49 Autenticación y despliegue de un subgrafo³⁵

Una vez desplegado el subgrafo, ya está disponible para realizar pruebas tanto en la propia página de Subgraph Studio como vía API como se puede ver en la [Figura 4.50](#).

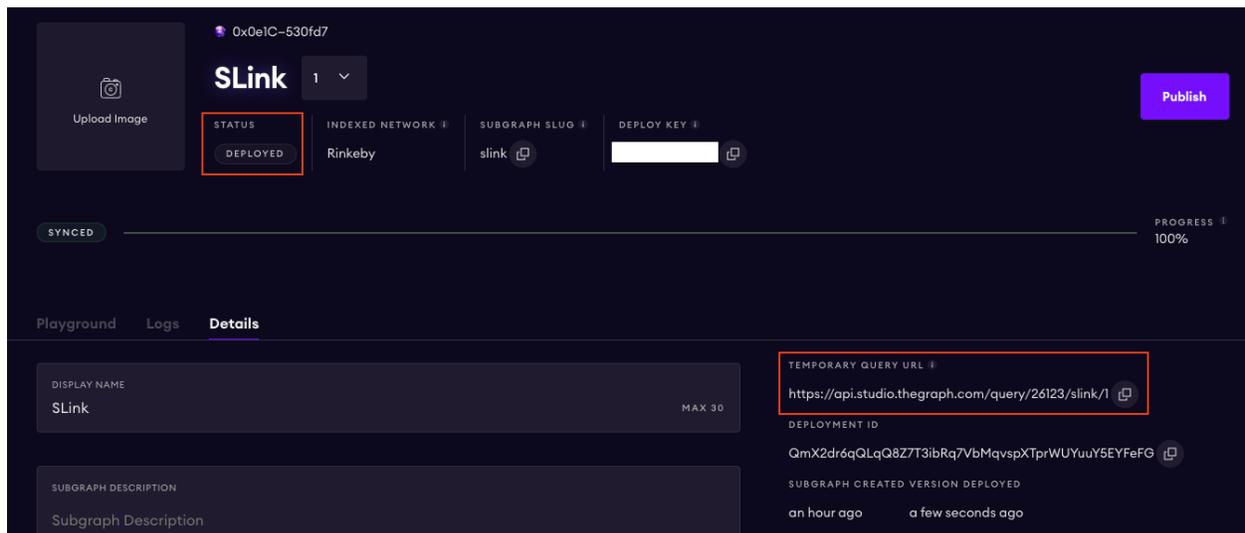


Figura 4.50 Página de edición de un subgrafo donde se muestran varios datos de este

Para finalizar el despliegue de forma correcta, se ha de publicar el Subgrafo. Para ello, se hace clic en el botón “Publish” y se configura cómo se desea realizar la publicación como se puede ver en la [Figura 4.51](#). Se recomienda marcar la opción “Ser el primero en señalar este subgrafo” para que se publique con mayor velocidad. Los subgrafos en The Graph deben ser seleccionados por los curadores para poder ser consultados y, de esta manera, saltamos el paso de esperar a que un curado seleccione el subgrafo. Esta opción no es gratuita y se debe

³⁴ <https://thegraph.com/docs/es/developer/quick-start/>

³⁵ <https://thegraph.com/docs/es/developer/quick-start/>

pagar con GRT, el token de The Graph. Cuanta más cantidad, más rápido se indexará. Para obtener tokens en la red de pruebas Rinkeby, se ha de visitar el servidor de Discord de The Graph y seguir los pasos indicados en este (<https://discord.gg/vtvv7FP>) en Discord³⁶.

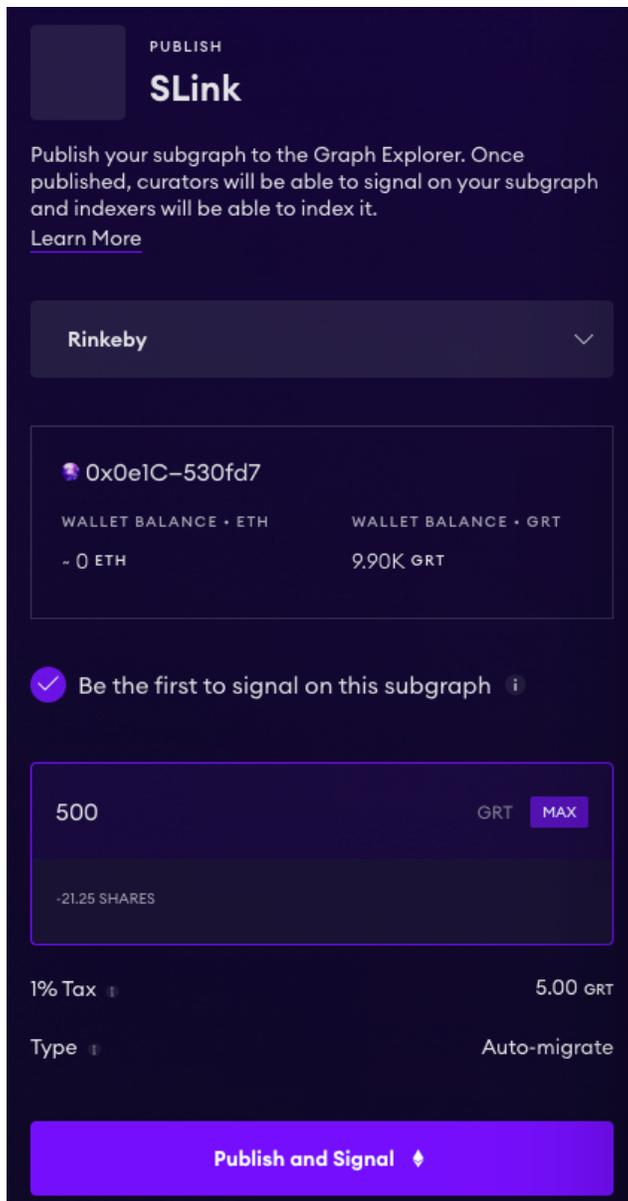


Figura 4.51 Configuración para la publicación de un subgrafo

Una vez el subgrafo está desplegado y publicado, se puede visitar en una dirección similar a <https://testnet.thegraph.com/subgraph?id=XX>, donde **XX** es el ID del subgrafo generado, aunque también se puede buscar en la página <https://testnet.thegraph.com/> para consultar toda

³⁶ <https://discord.com/>

la información disponible como se muestra en la [Figura 4.52](#). Estos enlaces son para la red de Rinkeby, en caso de ser en la red Ethereum principal, desaparece “*testnet*” de estos.

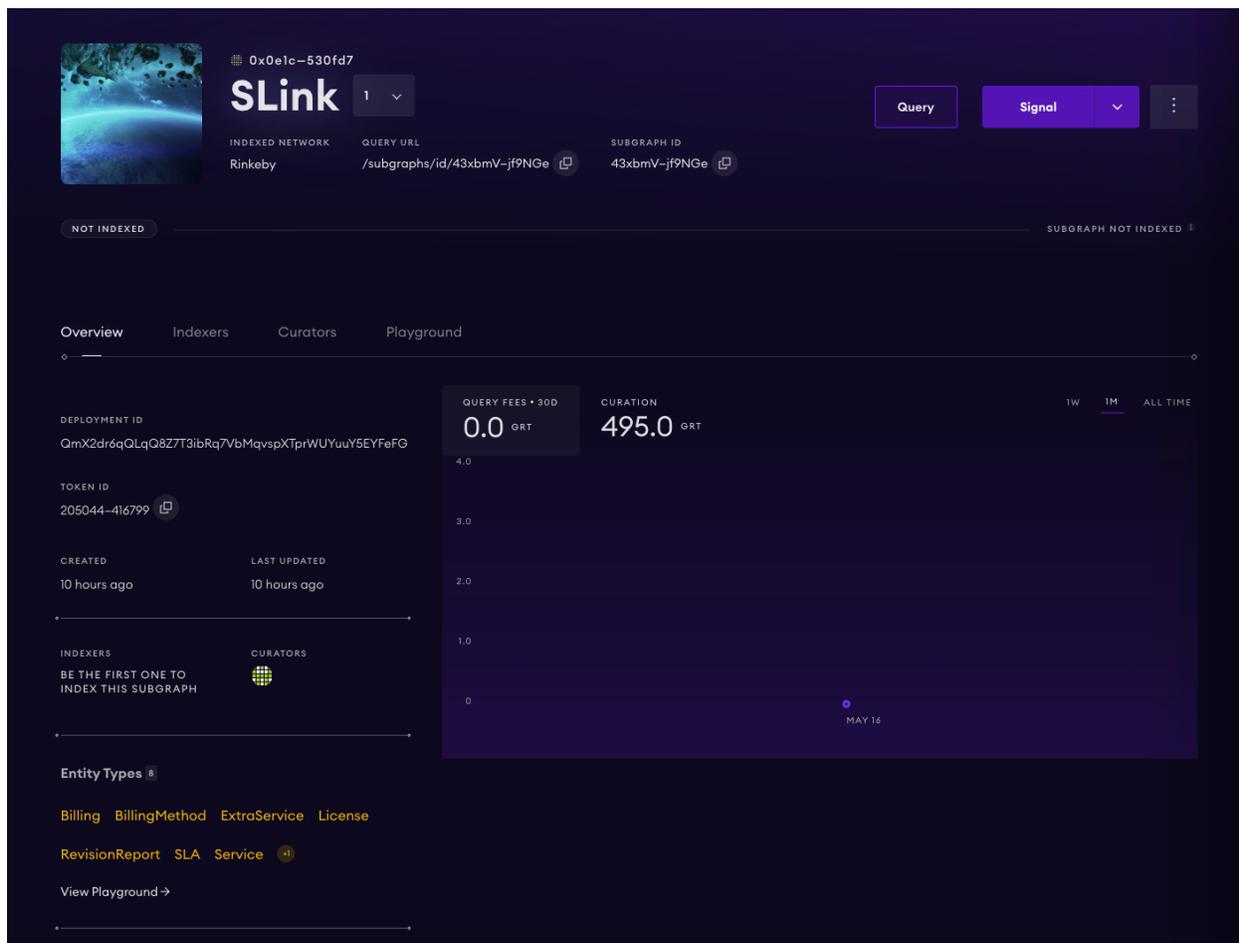


Figura 4.52 Subgrafo publicado a la espera de ser indexado

4.6.3. Despliegue del back-end en local

Los pasos a seguir para desplegar el back-end son:

1. Disponer de una instancia de MySQL en el sistema para posteriormente crear la base de datos privada dentro de ella.
2. Disponer de Node.js en el sistema para poder ejecutar aplicaciones Node.
3. Ejecutar el comando “*npm install*” desde una consola, dentro del directorio raíz del proyecto. Este comando instalará las dependencias del proyecto.
4. Ejecutar el comando “*node app.js*” desde una consola, dentro del directorio raíz del proyecto. Este comando iniciará el proyecto.

Todos estos pasos se encuentran en el fichero **Readme.md** en el repositorio del código fuente.

4.6.4. Despliegue del front-end en local

El front-end soporta tanto el idioma español como inglés, por lo que muchos campos de la configuración deben tener la traducción en ambos idiomas. Estos se marcan a continuación con la letra T.

Los pasos a seguir para desplegar el front-end son:

1. Indicar la dirección Ethereum propietaria del contrato inteligente en el fichero *config.js* del repositorio del código fuente.
2. Configuración de las opciones de un SLA (**disponible en el fichero *src/config.js* del repositorio del código fuente del front-end**):
 - a. **Servicios**: cada servicio debe contener un id, un nombre (T), una descripción (T), un precio, una periodicidad del precio y una indicación de si este es fijo u ocasional.
 - b. **Servicios extra**: misma configuración que los servicios.
 - c. **Espacios de servicio**: cada espacio de servicio debe contener un id, un nombre (T), una hora de inicio, una hora de fin, un precio, una periodicidad del precio y una indicación de si este es fijo u ocasional.
 - d. **Licencias**: cada licencia debe contener un id y un nombre (T).
 - e. **Periodicidades de informes de revisión**: cada periodicidad de informes de revisión debe contener un id, un precio, una periodicidad del precio y una indicación de si este es fijo u ocasional.
 - f. **Periodicidades de facturación**: cada periodicidad de facturación debe contener un id y una periodicidad.
 - g. **Métodos de facturación**: cada método de facturación debe contener un id y un nombre (T).
3. Disponer del back-end del sistema.
4. Disponer de Node.js en el sistema para poder ejecutar aplicaciones Node (imprescindible para el punto anterior).
5. Ejecutar el comando *"npm install"* desde una consola, dentro del directorio raíz del proyecto. Este comando instalará las dependencias del proyecto.

6. Ejecutar el comando `“cd yujo/”` desde una consola, dentro del directorio raíz del proyecto. Este comando cambiará la localización al código del proyecto.
7. Ejecutar el comando `“npm start”` desde una consola, dentro del directorio raíz del proyecto. Este comando iniciará el proyecto.

Todos estos pasos se encuentran en el fichero **Readme.md** en el repositorio del código fuente.

4.7. Licencia y código fuente

4.7.1. Licencia

En este proyecto se ha aplicado la siguiente licencia de Creative Commons³⁷ para el código fuente, salvo la parte de contrato inteligente (ver la [Figura 4.53](#)) Esto significa que se puede usar libremente el trabajo citando a los autores, no está permitido el uso comercial, modificaciones ni obras derivadas. Se puede observar dicha licencia en la parte superior de todas las páginas disponibles del sistema.



Figura 4.53 Licencia

Mientras para la parte de contrato inteligente, se ha utilizado la licencia MIT³⁸, licencia de software libre.

4.7.2. Código fuente

El código fuente de este proyecto está disponible en las siguientes direcciones de GitHub:

1. **Front-end:** <https://github.com/Yule1223/BC-ITSM.git>
2. **Back-end:** <https://github.com/JoseVelascoSantos/BC-ITSM-BE.git>

³⁷ <https://creativecommons.org/>

³⁸ https://es.wikipedia.org/wiki/Licencia_MIT

Capítulo 5 - Trabajo individual

En este capítulo se detalla el trabajo realizado por cada uno de los miembros del proyecto.

5.1. Trabajo individual de José Andrés

Realizar este TFG supuso al principio un reto debido al desconocimiento de muchas de las tecnologías que íbamos a utilizar. Pero antes de ponernos a desarrollar, mi compañera Yule y yo, tuvimos que realizar diversas investigaciones. Las primeras investigaciones fueron sobre ITIL, SLA, y la Blockchain de Ethereum, en ese mismo orden puesto que primero tendríamos que conocer ITIL para dar paso a los SLAs y a partir de ello, tendríamos el conocimiento sobre qué debe contener un SLA para tratar de averiguar cómo Solidity nos permitiría desarrollar un contrato inteligente para almacenarlos.

Una vez teníamos el conocimiento suficiente, empezamos el desarrollo. En mi caso, me ocupé del back-end aunque debido a mi experiencia profesional con React-Native (marco de trabajo para desarrollo multiplataforma de dispositivos móviles que nació a partir de React), ayudé a mi compañera con la arquitectura del front-end.

En primer lugar desarrollé una primera versión del contrato inteligente del sistema. Esta primera versión, daría lugar a realizar entonces un diseño de la base de datos para almacenar los datos sensibles. Teniendo esto, desarrollé la API REST con Express para juntar todo y permitir la creación de un SLA al completo.

El desarrollo del contrato inteligente se realizó de forma iterativa, donde en cada iteración nueva se actualizaba y mejoraba este hasta llegar a la versión final.

Al mismo tiempo que avanzaba en el diseño del contrato inteligente, pensamos que sería buena funcionalidad tener un listado de entidades para gestionar estas y desarrollé esa parte que implicaba actualizar la API REST y crear una nueva pantalla junto con mi compañera en front-end.

El dashboard fue lo último en desarrollar. Una vez pensamos en aquellos KPIs interesantes sobre un SLA, realicé una investigación y una prueba de concepto sobre The Graph. Surgieron dos problemas con ello: los eventos del contrato inteligente no eran suficientes para el subgrafo y la red de pruebas de The Graph no generaba momentáneamente tokens para poder publicar

el subgrafo. El primer problema se solventó actualizando el contrato inteligente y el segundo, con el paso de los días. Finalmente conseguí publicar el subgrafo que permitió ofrecer una API para realizar consultas sobre el contrato inteligente del sistema. Una vez se consiguió esto, actualicé la API REST del back-end para ofrecer un punto centralizado de consultas sobre todo el sistema y desarrollé la pantalla correspondiente a ello en front-end.

5.2. Trabajo individual de Yule

Debido a que este TFG abarca muchas tecnologías innovadoras y nunca vistas durante los estudios en la carrera, al principio de este TFG nos enfrentamos a muchos conocimientos desconocidos. Por ello, nos pusimos primero a realizar investigaciones sobre ITIL y SLA. En mi caso, me puse a leer libros sobre ITIL para estudiar los campos que pueden tener un SLA. Gracias a nuestra directora Maricruz que nos ofreció muchos libros respecto a ello y nos facilitó mucho el proceso de investigaciones.

Una vez teníamos asentados los campos que puede tener un SLA, a partir de la arquitectura que realizó mi compañero José Andrés, me puse con la parte front-end del sistema, diseñando las siguientes páginas:

- Página principal
- Página de SLA
- Página de Conocernos
- Página de Aviso legal y política de privacidad

Nos reunimos con nuestra directora Maricruz para ver la mejor forma en la que se queda cada página. Al principio la página SLA era con paginaciones, estando una página para los datos del cliente, una para la empresa y otra para SLA, al final realicé cambios juntando las 3 en una misma, como se puede observar en las figuras de la página de SLA en el capítulo 3. Hice bastantes modificaciones en la página de SLA hablando con nuestra directora Maricruz para ver qué campos son más necesarios y comunes para un SLA.

También me encargué de implementar el cálculo automático de precio total de SLA, colocado en la parte inferior de la página de SLA. De manera que se pueda mostrar un precio de referencia a base de las opciones elegidas por el cliente, aunque el precio real se podría variar en función de sus necesidades.

Otro paso muy importante para este proyecto es definir los avisos legales y la política de privacidad, por ello, me puse a investigar la información que deberíamos aclarar a los clientes en el momento de realizar una petición de contacto o enviar un SLA a través del sistema. Me encargué de definir los elementos legales que pueden afectar al sistema.

En los pasos finales del sistema, me encargué de realizar ajustes y mejoras en front-end para que las páginas se vean más intuitivas.

Para ir completando la memoria, me encargué de redactar los capítulos de resumen, agradecimientos, introducción, estado de cuestión, y trabajé junto con mi compañero José Andrés para redactar el capítulo del sistema SLink.

Capítulo 6 - Conclusiones y trabajo futuro

A lo largo de este capítulo se presentan las conclusiones obtenidas en este TFG y el trabajo futuro que se podría realizar si se continuara el desarrollo de este proyecto.

6.1. Conclusiones

Las principales conclusiones que obtenemos fruto del trabajo realizado para este TFG están relacionadas con la integración de los SLAs dentro de la Blockchain de Ethereum.

Hemos visto que ofrece transparencia al tratarse de una red donde cualquier interesado externo puede informarse sobre los SLAs creados y registrados dentro de la misma, garantizando una trazabilidad e inmutabilidad que quizás solo una red de este tipo pueda asegurar por ser una red descentralizada. El uso de esta red no es impedimento para el sistema SLink, pues si bien es verdad que las transacciones duran unos segundos con el protocolo actual de consenso Prueba de Trabajo, este tiempo es aceptable y el resto de tiempos donde se ve implicada la red (consultas a la red Blockchain vía The Graph) son muy cortos.

The Graph, por cómo funciona (ver la sección [3.13.2. ¿Cómo funciona The Graph?](#)), tiene los datos disponibles a ser consultados ya listos, por lo que la velocidad de consulta no se ve comprometida. Además, The Graph permite hacer todo tipo de consultas con numerosas opciones de configuración para estas, proporcionando esto una gran flexibilidad a la hora de realizar búsquedas de datos sobre el contrato inteligente. Esto, junto con un modelo de datos apropiado gracias a los eventos del contrato inteligente, permite tener todo lo necesario para un dashboard completo.

La aceptación en el mundo empresarial de certificaciones obtenidas a través de la Blockchain es ya una realidad: Iberdrola³⁹ (empresa española dedicada a la producción, distribución y comercialización de energía), recientemente ha utilizado esta tecnología para verificación de votos⁴⁰. Además, a finales de diciembre de 2021 una Sentencia de la Audiencia Provincial de

³⁹ <https://www.iberdrola.es/>

⁴⁰

<https://www.ciospain.es/seguridad/la-tecnologia-blockchain-se-cuela-en-la-junta-de-accionistas-de-iberdrola>

Vitoria⁴¹ admitió el uso de Blockchain como elemento tecnológico que permite una mínima auditoría de autenticidad para un documento electrónico presentado como prueba en el procedimiento. Ambos hechos transmiten que la certificación y el reconocimiento de contratos inteligentes establecidos y registrados en la Blockchain es cada vez más reconocida.

Por otro lado, en lo que respecta al desarrollo, Javascript es una tecnología para aplicaciones muy versátil que se debería de tener en cuenta para un desarrollo ágil. Como se ha visto en este proyecto, Javascript nos ha permitido trabajar con un marco de trabajo front-end (React) y con un marco de trabajo back-end (Express). Fuera de este proyecto, se pueden desarrollar aplicaciones móviles multiplataforma (React-Native por ejemplo⁴²) y aplicaciones de escritorio (Electron por ejemplo⁴³). Por si esto fuera poco, la sintaxis de lenguajes como Solidity⁴⁴, se asemejan bastante, lo cual es una gran ayuda a la hora de aprender otra tecnología.

Creemos también que la adopción ITIL debe aumentar en relación a los servicios de TI ofrecidos en la actualidad por las empresas debido al crecimiento del sector en los últimos años y lo que pueda crecer en el futuro. A pesar de ser utilizada por cientos de organizaciones, es necesario aplicar las buenas prácticas de ITIL en muchas más para conseguir una prestación de servicios de calidad en una sociedad cada vez más digitalizada; una empresa no solo debe digitalizarse para ofrecer servicios de TI, sino que estos han de ser de calidad.

Igualmente, los estudiantes universitarios vinculados a cualquier rama de la informática deberían estar familiarizados con los conocimientos de SLA, dado que es una parte fundamental en los contratos de una empresa del sector de TI. Conocer las características y aplicaciones de los SLAs en el mundo empresarial ayudaría a mejorar sus carreras profesionales.

Por último, creemos que la tecnología Blockchain tiene un gran futuro hasta el punto de quizás revolucionar completamente el sector de TI, tal y como se conoce en la actualidad. La llegada de Web3, una nueva iteración sobre la web con tecnología Blockchain, incorpora todos los conceptos de la tecnología Blockchain como la descentralización y economía digital. Pero esta tecnología no se queda solo en eso: permite la existencia de las criptomonedas (que siguen

⁴¹ <https://www.poderjudicial.es/search/AN/openDocument/954cf088a7a89e01/20220510>

⁴²

<https://www.izertis.com/es/-/blog/react-native-de-javascript-al-desarrollo-de-aplicaciones-moviles>

⁴³ <https://www.electronjs.org/>

⁴⁴ https://www.tutorialspoint.com/solidity/solidity_basic_syntax.htm

creciendo con el paso del tiempo y la creación de nuevos proyectos, permiten fomentar todo lo que esté relacionado con Blockchain), NFTs (del inglés *Non Fungible Tokens*, en castellano Tokens No Fungibles) (permiten registrar en una Blockchain una propiedad digital) y Metaversos⁴⁵ (mundos digitales donde la mayoría utilizan tecnología Blockchain).

El objetivo de este TFG era crear un sistema descentralizado basado en tecnología Blockchain para favorecer la gestión de SLAs entre proveedores de servicios de TI y clientes siguiendo las buenas prácticas ITIL y creemos que esto se ha conseguido. A pesar de no haber puesto en un entorno de producción a SLink, creemos que se trata de un sistema que consigue lo anterior.

Gracias a este TFG, hemos obtenido conocimiento de numerosas tecnologías:

- **React** como marco de trabajo front-end
- **Postman** como aplicación de pruebas para APIs REST
- **Ethereum** como red Blockchain destinada a aplicaciones
- **Solidity** como lenguaje de contratos inteligentes
- **Remix** como entorno de desarrollo y despliegue de contratos inteligentes de la red Ethereum
- **GraphQL** como lenguaje de consulta y manipulación de datos para APIs
- **The Graph** como motor de búsqueda para contratos inteligentes
- El resto de tecnologías ya se conocían por el conocimiento adquirido en **asignaturas del grado de ingeniería del software en la facultad informática de la Universidad Complutense de Madrid**:
 - **Bases de datos (BD)**: En esta asignatura nos dieron la introducción de las bases de datos relacionales, nos enseñaron a realizar las consultas a la base de datos.
 - **Aplicaciones Web (AW)**: En esta asignatura nos enseñaron los conocimientos sobre HTML, CSS, Javascript y Express usados en el desarrollo del front-end del sistema. También aprendimos el manejo de servicio web y la conexión entre los mismos.
 - **Gestión de Proyectos Software (GPS)**: En esta asignatura nos enseñaron utilizar las herramientas del control de versiones, trabajar en equipo, aprender a redactar documentos formales, y sobre todo, fue donde conocimos a nuestra directora del TFG y empezó la buena relación entre nosotros.

⁴⁵ <https://es.wikipedia.org/wiki/Metaverso>

- **Ética, Legislación y Profesión (ELP):** En esta asignatura nos informaron el uso de las licencias, elegir una licencia adecuada para proteger nuestro trabajo. También nos explicaron sobre el software libre y la propiedad de TFG.

6.2. Trabajo futuro

El sistema SLink tiene numerosas posibilidades de ampliación que, debido a la limitación de tiempo y el fin de estudio, no se han llegado a realizar. A continuación, se mencionan algunas posibilidades de ampliación que se podrían llevar a cabo en un futuro:

- **Pago del SLA en el propio sistema:** Si bien existe el pago de una tasa por parte del cliente para crear un nuevo SLA en el contrato inteligente de la red Ethereum (pago necesario para pagar la transacción que introduce cambios en la red), sería útil tener métodos de pago dentro del propio sistema. Por ejemplo, si el método de facturación elegido es transferencia bancaria, se podría proporcionar los datos bancarios necesarios para ello e incluso pedir un justificante de la transferencia. Si el método de facturación elegido es el pago con criptomonedas, se podría modificar el contrato inteligente para admitir esta funcionalidad y realizar el pago junto a la tasa de la transacción.
- **Gestión documental:** Sería interesante una gestión documental para SLAs, de manera que cada SLA pudiera tener una serie de documentos proporcionados bien por el proveedor de servicios o por el cliente. Esto haría posible, por ejemplo, la descarga de los niveles de servicio ofertados en un documento o la subida de una documentación concreta pedida al cliente por el proveedor de servicios.
- **Resumen SLA en forma de documento:** Esta funcionalidad haría posible la obtención en forma de documento de un resumen del SLA creado por parte del cliente y reduciría la complejidad de revisar este SLA en el contrato inteligente de forma externa al sistema.
- **Más opciones de monedero digital:** Actualmente el sistema solamente dispone una opción de monedero digital, MetaMask, sería interesante si dispone de más opciones para llegar al mayor público.
- **Darse de alta un cliente vía front-end:** Actualmente si un cliente quiere darse de alta, tiene que enviar una petición de contrato a través de la página de Conocernos (ver la sección [4.3.9. Página de Conocernos](#)), lo ideal es tener una página donde el cliente

pueda darse de alta por su cuenta. De esta manera, el cliente ahorra el tiempo de espera y los personales de SLink ahorra la carga de trabajo.

- **Mostrar mensajes de error con llamadas API:** Actualmente no el sistema no dispone de mensajes de error con llamadas API, sería muy interesante tenerlo.
- **Roles y permisos de la aplicación:** La existencia de roles y permisos de la aplicación permitiría tener más de un propietario (de manera que más de una dirección Ethereum podría visualizar la lista de entidades y el Dashboard) o la existencia de usuarios con permisos para ello.
- **Despliegue del sistema vía front-end:** Actualmente el sistema se despliega para un entorno de producción de forma manual. Si existiera un despliegue en front-end, de manera que se siguieran los pasos necesarios sin requerir un conocimiento avanzado por parte del usuario, SLink podría ser desplegado sin ayuda y eso aumentaría la viabilidad del sistema.
- **Configuración del sistema vía front-end:** En la versión actual del sistema, se debe configurar este por un archivo de configuración (src/config.js en el repositorio del código fuente del front-end). Sería interesante tener la posibilidad de realizar esta configuración desde el front-end y/o mostrar la configuración actual en cualquier momento.
- **Pruebas automatizadas de toda la aplicación:** Por ejemplo, realización de pruebas unitarias para los componentes React creados y puntos finales de la API REST del back-end, pruebas funcionales para las pantallas del front-end y pruebas de rendimiento del sistema en general.
- **Chat en directo como método de contacto:** Esta mejora supone la existencia de personal pendiente de atender a clientes o bien un robot capaz de guiar al cliente hacia la ayuda necesaria para su necesidad. Sería útil para dar de alta a nuevos clientes de una forma rápida y actual junto a la existente opción de llamada telefónica (también opción de contacto rápido).
- **Definir KPI más apropiados según la necesidad de las organizaciones:** Dado la limitación de tiempo y alcance de este proyecto, se ha tenido que definir KPIs sencillos siendo un prototipo. Se tendrá que definir KPI más complejas según la necesidades de cada empresa en el entorno real.

Chapter 6 - Conclusions and future work

This chapter presents the conclusions obtained in this Final Degree Project (FDP) and the future work that could be carried out if the development of this project were continued.

6.1. Conclusions

The main conclusions that we obtain as a result of the work carried out for this FDP are related to the integration of SLAs within the Ethereum Blockchain.

We have seen that it offers transparency since it is a network where any interested external party can find out about the SLAs created and registered within it, guaranteeing traceability and immutability that perhaps only a network of this type can ensure as it is a decentralized network. The use of this network is not an impediment for the system SLink, because although it is true that transactions last a few seconds with the current Proof-of-Work consensus protocol, this time is acceptable and the rest of the times where the network is involved (consultations to the Blockchain network via The Graph) are very short.

The Graph, because of how it works (see section [3.13.2. ¿Cómo funciona The Graph?](#)), has the available data to be queried ready, so query speed is not compromised. In addition, The Graph allows all kinds of queries with numerous configuration options for them, providing great flexibility when searching for data on the smart contract. This, together with an appropriate data model thanks to the smart contract events, allows you to have everything you need for a complete dashboard.

The acceptance in the business world of certifications obtained through the Blockchain is already a reality: Iberdrola⁴⁶ (a Spanish company dedicated to the production, distribution and sale of energy), has recently used this technology to verify votes⁴⁷. In addition, at the end of December 2021, a Judgment of the Provincial Court of Vitoria⁴⁸ admitted the use of Blockchain as a technological element that allows a minimum authenticity audit for an electronic document

⁴⁶ <https://www.iberdrola.es/>

⁴⁷

<https://www.ciospain.es/seguridad/la-tecnologia-blockchain-se-cuela-en-la-junta-de-accionistas-de-iberdrola>

⁴⁸ <https://www.poderjudicial.es/search/AN/openDocument/954cf088a7a89e01/20220510>

presented as evidence in the procedure. Both facts convey that the certification and recognition of smart contracts established and registered in the Blockchain is increasingly recognized.

On the other hand, when it comes to development, Javascript is a very versatile application technology that should be considered for agile development. As seen in this project, Javascript has allowed us to work with a front-end framework (React) and a back-end framework (Express). Outside of this project, cross-platform mobile applications (React-Native for example⁴⁹) and desktop applications (Electron for example⁵⁰) can be developed. As if this were not enough, the syntax of languages like Solidity⁵¹ are quite similar, which is a great help when learning another technology.

We also believe that ITIL adoption should increase in relation to the IT services currently offered by companies due to the growth of the sector in recent years and what may grow in the future. Despite being used by hundreds of organizations, it is necessary to apply ITIL good practices in many more to achieve quality service provision in an increasingly digitized society; A company should not only be digitized to offer IT services, but these should also be of quality.

Similarly, university students linked to any branch of computing should be familiar with the knowledge of SLA, since it is a fundamental part of the contracts of a company in the IT sector. Knowing the characteristics and applications of SLAs in the business world would help improve their professional careers.

Finally, we believe that Blockchain technology has a great future to the point of perhaps completely revolutionizing the IT sector as it is known today. The arrival of Web3, a new iteration on the web with Blockchain technology, incorporates all the concepts of Blockchain technology such as decentralization and digital economy. But this technology is not just that: it allows the existence of cryptocurrencies (which continue to grow over time and the creation of new projects, they allow everything related to Blockchain to be promoted), NFTs (Non Fungible Tokens) (allow a digital property to be registered in a Blockchain) and Metaverses⁵² (digital worlds where most use Blockchain technology).

The objective of this FDP was to create a decentralized system based on Blockchain technology to favor the management of SLAs between IT service providers and clients following ITIL good

⁴⁹ <https://reactnative.dev/docs/javascript-environment>

⁵⁰ <https://www.electronjs.org/>

⁵¹ https://www.tutorialspoint.com/solidity/solidity_basic_syntax.htm

⁵² <https://en.wikipedia.org/wiki/Metaverse>

practices and we believe that this has been achieved. Despite not having put SLink into a production environment, we believe that it is a system that achieves the above.

Thanks to this FDP, we have obtained knowledge of numerous technologies:

- **React** as a front-end framework
- **Postman** as a testing application for REST APIs
- **Ethereum** as a Blockchain network intended for applications
- **Solidity** as a smart contract language
- **Remix** as a development and deployment environment for smart contracts on the Ethereum network
- **GraphQL** as a query and data manipulation language for APIs
- **The Graph** as a search engine for smart contracts
- The rest of the technologies were already known from the knowledge acquired in subjects of the **software engineering degree in the computer science faculty of the Complutense University of Madrid**:
 - **Databases (BD)**: In this subject we were given the introduction of relational databases, and we were taught to make queries to the database.
 - **Web Applications (AW)**: In this subject we were taught the knowledge about HTML, CSS, Javascript, and Express, which are used in the development of the front-end of the system SLink. We also learned the management of web services and the connection between them.
 - **Software Project Management (GPS)**: In this subject we were taught to use version control tools, work as a team, learn to write formal documents, and above all, it was where we met our director of the FDP and the good relationship between us began.
 - **Ethics, Legislation and Profession (ELP)**: In this course we were informed about the use of licenses, choosing an appropriate license to protect our work. They also explained to us about free software and the property of FDP.

6.2. Future work

The SLink system has numerous expansion possibilities that, due to time limitations and the end of the study, have not been carried out. Here are some expansion possibilities that could be carried out in the future:

- **Payment of the SLA in the system itself:** Although there is the payment of a fee by the client to create a new SLA in the smart contract of the Ethereum network (payment necessary to pay for the transaction that introduces changes in the network), it would be useful have payment methods within the system itself. For example, if the chosen billing method is bank transfer, you could provide the necessary bank details for it and even request a transfer receipt. If the chosen billing method is cryptocurrency payment, the smart contract could be modified to support this functionality and make the payment together with the transaction fee.
- **Document management:** Document management for SLAs would be interesting, so that each SLA could have a series of documents provided either by the service provider or by the client. This would make it possible, for example, to download the service levels offered in a document or to upload specific documentation requested from the client by the service provider.
- **SLA summary in the form of a document:** This functionality would make it possible to obtain a summary of the SLA created by the client in the form of a document and would reduce the complexity of reviewing this SLA in the smart contract externally to the system.
- **More digital wallet options:** Currently the system only has one digital wallet option, MetaMask, it would be interesting if it had more options to reach the largest audience.
- **Register a client via front-end:** Currently, if a client wants to register, they have to send a contract request through the Know Us page (see section [4.3.9. Página de Conocernos](#)), the ideal is to have a page where the client can register by themselves. In this way, the customer saves the waiting time and the SLink personnel saves the workload.
- **Show error messages with API calls:** Currently the system does not have error messages with API calls, it would be very interesting to have it.
- **Roles and permissions of the application:** The existence of roles and permissions of the application would allow it to have more than one owner (so that more than one Ethereum address could view the list of entities and the Dashboard) or the existence of users with permissions to do so.
- **System deployment via front-end:** Currently the system is deployed to a production environment manually. If there was a front-end deployment, so that the necessary steps

were followed without requiring advanced knowledge on the part of the user, SLink could be deployed without help and that would increase the viability of the system as well.

- **System configuration via front-end:** In the current version of the system, it must be configured through a configuration file (src/config.js in the front-end source code repository). It would be interesting to have the possibility to make this configuration from the front-end and/or show the current configuration at any time.
- **Automated testing of the entire application:** For example, unit testing for built React components and back-end REST API endpoints, functional testing for front-end screens, and overall system performance testing.
- **Live chat as a contact method:** This improvement supposes the existence of staff waiting to attend to clients or a robot capable of guiding the client towards the necessary help for their need. It would be useful to register new clients in a fast and current way, together with the existing telephone call option (also a quick contact option).
- **Define the most appropriate KPIs according to the needs of the organizations:** Given the limited time and scope of this project, simple KPIs had to be defined as a prototype. More complex KPIs will have to be defined according to the needs of each company in the real environment.

Bibliografía

1. Ana Andrés Álvarez, Carlos Manuel Fernández Sánchez y Boris Delgado Riss. (2A. ED.). *GUÍA PRÁCTICA DE ISO/IEC 20000-1 PARA SERVICIOS TIC*. Universidad Complutense de Madrid. <https://elibro.net/es/ereader/universidadcomplutense/131803> [Último acceso: 16/5/2022].
2. Jan van Bon (2008). *Fundamentos de la Gestión de Servicios de TI basada en ITIL*. Van Haren Publishing. <https://books.google.com.pe/books?id=WdFEBAAAQBAJ&printsec=copyright#v=onepage&q&f=false> [Último acceso: 20/5/2022]
3. Rodríguez, J. R. (1 de marzo de 2022). *¿Hace falta un departamento de informática? Pues, depende*. Tecnología++. Blogs Institucionals UOC. <https://blogs.uoc.edu/informatica/departamento-informatica/> [Último acceso: 16/5/2022]
4. OLCESE, A., & Moya, C. (3 de Febrero de 2022). *El tejido productivo no se recupera: España tiene hoy 77.831 empresas menos que antes de la pandemia*. El Mundo. <https://www.elmundo.es/economia/macroeconomia/2022/02/03/61fab040fc6c832f698b45eb.html> [Último acceso: 16/5/2022]
5. González Fernández-Villavicencio, N.; Menéndez Novoa, J.L.; Seoane García, C.; San Millán Fernández, M.E. (2013). *Revisión y propuesta de indicadores (KPI) de la Biblioteca en los medios sociales*. Revista Española de Documentación Científica, 36(1):e005. <http://dx.doi.org/10.3989/redc.2013.1.919>. [Último acceso: 16/5/2022]
6. IBM Blockchain. (n.d.). *What is Blockchain Technology?*. IBM. <https://www.ibm.com/es-es/topics/what-is-blockchain> [Último acceso: 16/5/2022]
7. Vitalik Buterin (2014). *Ethereum White Paper*. <https://ethereum.org/en/whitepaper/> [Último acceso: 24/5/2022]
8. Sinclair Davidson, Primavera De Filippi, Jason Potts (2016). *Economics of Blockchain*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2744751#:~:text=The%20basic%20economics%20of%20blockchain.cost%20of%20processing%20digital%20information%2C [Último acceso: 23/5/2022]
9. Alex Banks & Eve Porcello (2017). *Learning React: Functional Web Development with React and Redux*. <https://media.graphcms.com/HrM5QEgWSweYEQBwCISG?dl=true> [Último acceso: 23/5/2022]

10. Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*.
<https://bitcoin.org/bitcoin.pdf> [Último acceso: 24/5/2022]
11. Swan, M. (2015). *Blockchain: blueprint for a new economy*. First edition. ed. O'Reilly, Beijing : Sebastopol, CA.
<http://book.itep.ru/depository/blockchain/blockchain-by-melanie-swan.pdf> [Último acceso: 24/5/2022]
12. Steinberg, R. A. (2006). *Measuring ITIL: Measuring, Reporting and Modeling - The IT Service Management Metrics That Matter Most to IT Senior Executives*. Trafford Publishing
<https://www.scribd.com/book/387796481/Measuring-Itsm-Measuring-Reporting-and-Modeling-the-It-Service-Management-Metrics-That-Matter-Most-to-It-Senior-Executives>
[Último acceso: 25/5/2022]
13. Antonopoulos, A.M.;Wood, G. (2019). *Mastering Ethereum: building smart contracts and DApps*. First edition. ed. O'Reilly, Sebastopol, CA.
https://dl.ebooksworld.ir/motoman/Mastering_Ethereum_Andreas.M.Antonopoulos.www.EBooksWorld.ir.pdf [Último acceso: 25/5/2022]
14. Julija Golosova; Andrejs Romanovs. (2018). *The Advantages and Disadvantages of the Blockchain Technology*. IEEE. <https://ieeexplore.ieee.org/document/8592253> [Último acceso: 25/5/2022]

