UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

ContractMe

Presentado por David Martínez Bahillo
en Universidad de Burgos — 28 de febrero de 2024
Tutor: Sandra Rodriguez Arribas

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. David Martínez Bahillo, con DNI 71566321G, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado «ContractMe».

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 28 de febrero de 2024

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
D. nombre tutor	D. nombre co-tutor

Resumen

Este proyecto se enfoca en la creación de una solución innovadora para la contratación laboral, aprovechando la tecnología blockchain y los contratos inteligentes. Se busca agilizar el proceso de contratación en sectores como la agricultura y servicios domésticos mediante la automatización de la gestión de contratos desde su inició hasta su final. La solución propuesta incluye funcionalidades avanzadas como el seguimiento del tiempo de trabajo y el pago automático, integrando tecnologías de localización GPS para establecer zonas de trabajo específicas. El proyecto incorpora métodos de autenticación biométrica y escaneo de códigos QR, apoyándose en soluciones de identidad descentralizadas para proteger la privacidad de los usuarios.

El objetivo final es proporcionar una herramienta que promueva prácticas de empleo transparentes y legales, optimizando los procesos de contratación y verificación para empleadores y trabajadores garantizando la seguridad de todos los tramites.

Descriptores

Blockchain, contratos inteligentes, localización GPS, autenticación biométrica de teléfonos inteligentes, encriptación de datos, desarrollo de aplicación móvil, procesamiento de pagos.

Abstract

This project focuses on creating an innovative solution for employment hiring, leveraging blockchain technology and smart contracts. It aims to streamline the hiring process in sectors such as agriculture and domestic services by automating contract management from start to finish. The proposed solution includes advanced features such as work time tracking and automatic payment, integrating GPS technology to establish specific work zones. The project incorporates biometric authentication methods and QR code scanning, relying on decentralized identity solutions to protect user privacy.

The final goal is to provide a tool that promotes transparent and legal employment practices, optimizing hiring and verification processes for employers and workers, ensuring the security of all procedures.

Keywords

Blockchain, smart contracts, GPS tracking, smartphone biometric authentication, data encryption, mobile app development, payment processing.

Índice general

Índice de figuras

Índice de tablas

1. Introducción

La transformación digital se ha convertido en un pilar fundamental para el desarrollo y la eficiencia de diversos sectores económicos, entre ellos el mercado laboral. Sin embargo, a pesar de los avances tecnológicos, ciertos sectores como la agricultura, los servicios domésticos y el trabajo freelance enfrentan retos significativos en la gestión eficiente de contrataciones y la verificación de identidad y tiempo laboral. Estos desafíos se ven agravados por la presencia de la economía sumergida, donde la falta de transparencia y la informalidad laboral no solo perjudican la economía global, sino que también vulneran los derechos de los trabajadores. En este contexto, se pretende implementar una solución innovadora que permita superar las barreras existentes, haciendo frente a la economía sumergida mediante el empleo de tecnologías disruptivas como blockchain y contratos inteligentes. Esta iniciativa busca promover la formalización de empleos y asegurar prácticas laborales justas, abordando directamente los problemas de falta de transparencia y seguridad en los procesos contractuales. Se destacará el impacto social y económico de reducir la economía sumergida, apoyándose en datos que evidencian su magnitud y las implicaciones positivas de una mayor formalización laboral.

La adopción de contratos inteligentes en una red blockchain permite una gestión contractual transparente, segura y automática, asegurando que los términos acordados se cumplan eficazmente sin intervención manual. Esta tecnología también proporciona un registro inmutable y verificable de las transacciones y acuerdos labores, combatiendo así prácticas ilegales y fomentando un entorno de trabajo más equitativo.

Este proyecto no solo aborda la necesidad de un sistema de contratación y verificación más eficiente y seguro sino que también se anticipa a las demandas de un mercado laboral en constante evolución, ofreciendo una

herramienta adaptable y escalable. Con una interfaz de usuario diseñada para la simplicidad y la accesibilidad en dispositivos móviles, se busca fomentar prácticas de empleo legales y transparentes, contribuyendo a la creación de un entorno laboral más justo y seguro para todos. La inclusión de tecnologías GPS y la autenticación biométrica refuerza este objetivo, permitiendo un seguimiento preciso de las horas de trabajo y asegurando la identidad de los trabajadores de manera segura y privada.

Por tanto este proyecto no solo propone una solución práctica que beneficia tanto a trabajadores como empleadores, sino que también tiene el potencial de impactar positivamente en la economía global. marcando un paso adelante hacia la erradicación de la economía sumergida y el establecimiento de prácticas laborales más justas y transparentes a nivel mundial.

2. Objetivos del proyecto

En esta sección se muestran las metas que se pretenden alcanzar con el desarrollo del proyecto.

Objetivos técnicos

1. **Seguir estándares en el desarrollo de contratos inteligentes**: Alineándose con las mejores prácticas de la comunidad Ethereum y Solidity.
2. **Adoptar metodologías ágiles**: Para favorecer un desarrollo iterativo y eficiente, facilitando la adaptación a cambios y mejora continua del proyecto.
3. **Documentar del proyecto**: Utilizando herramientas visuales para explicar la arquitectura.
4. **Mantener un alto estándar de calidad en el código**: Empleando herramientas de revisión de código.
5. **Evaluar la calidad de la solución**: En términos de eficiencia.
6. **Planear un plan de implementeción y escalabilidad**

Objetivos del Software

1. **Integrar Ethereum en el proyecto**: Asegurando una implementación efectiva para el despliegue de contratos inteligentes

2. **Desarrollar contratos inteligentes con Solidity**: Asegurando un código eficiente y adaptado a las necesidades específicas de la contratación laboral.

3. **Desplegar en una red Ethereum real**: Logrando una convivencia con el resto de contratos de la comunidad.

4. **Implementar tecnologías de autenticación biométrica**: Utilizando métodos como la huella dactilar o el reconocimiento facial para la identificación del usuario.

5. **Integrar tecnología GPS**: Verificando la presencia del trabajador en el lugar adecuado y el momento correcto.

6. **implementar códigos QR**: Agilizando los procesos administrativos y operativos además de aplicar una capa adicional de seguridad.

7. **Integrar servicios de inicio sesión**: Proporcionando a los usuarios opciones de inicio de sesión descentralizadas a través de MetaMask.

8. **Desarrollar una interfaz de usuario intuitiva y accesible**: Priorizando la simplicidad y la usabilidad para garantizar una plataforma fácil de utilizar para todos los usuarios.

3. Conceptos teóricos

A continuación se sintetizarán algunos conceptos teóricos relevantes para la correcta compresión del proyecto.

3.1. Introducción a la Blockchain

La Blockchain proporcionando un registro inmutable y en tiempo real de transacciones, emergió como una solución al problema de doble gasto [9] en transacciones digitales, un desafío que había eludido a los criptógrafos durante décadas. En 2008 una persona o grupo anónimo bajo el seudónimo de Satoshi Nakamoto publicó un documento técnico para crear una moneda digital para contabilizar y transferir valor. Así nació una tecnología que se fundamenta en una base de datos descentralizada compuesta por bloques de información replicados y sincronizados en múltiples ordenadores. Esta premisa hizo que en enero de 2009 entrará en funcionamiento la primera red basada en el protocolo Bitcoin [8]. La BlockChain demostró su capacidad para contabilizar y transferir valor de manera segura sin la necesidad de intermediarios, transformando el sector financiero y encontrando aplicaciones en una variedad de industrias.

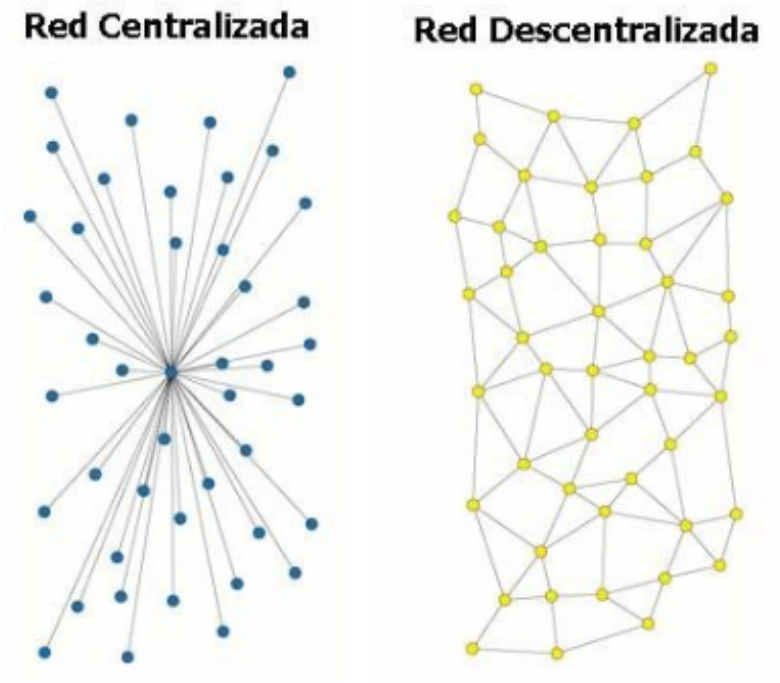

Figura 3.1: Diferencias entre una red centralidad y descentralizada

Tecnología subyacente

La tecnología Blockchain funciona como un libro mayor digital (DTL) que registra transacciones en múltiples ordenadores de manera que cada registro es inalterable e irreversible. Este registro se organiza en bloques de datos que están interconectados de manera cronológica formando una cadena [16].

Cada bloque contiene un numero determinado de transacciones y esta formado por tres elementos principales; el dato de la transacción, el hash del bloque anterior en la cadena (lo que asegura la continuidad de la cadena) y su propio hash único, generado a partir de la información contenida en el bloque. El hash es una función criptográfica que produce una salida de longitud fija a partir de una entrada de longitud variable, cualquier cambio en la información del bloque alteraría drásticamente su hash, evidenciando cualquier intento de fraude [3].

El proceso de añadir un nuevo bloque a la cadena requiere de un consenso entre los nodos de la red, lo cual se logra mediante diferentes mecanismos, siendo el "Proof of Work"(Prueba de trabajo) uno de los más utilizados

ocupando alrededor del 60 % de la capitalización total. [12] Este mecanismo implica resolver un problema criptográfico complejo que requiere un gran poder de computo. Funciona como una competición, en la que el nodo más rápido en resolver el problema obtiene el derecho de añadir el nuevo bloque a la cadena y es recompensado, generalmente con criptomonedas.

Una vez un bloque es añadido a la cadena, se distribuye a todos los nodos de la red, actualizando así el libro mayor en cada nodo. Esta distribución asegura una redundancia que la hace segura contra la manipulación, ya que alterar un registro requeriría cambiar el bloque correspondiente y todos los bloques posteriores en la mayoría de los nodos de la red, una tarea casi imposible debido a la demanda de computo.

Utilidades de la Blockchain

Su aplicación más reconocida ha sido Bitcoin, ha demostrado las capacidades de la Blockchain, permitiendo transacciones globales rápidas y seguras, caracterizadas por su alta liquidez, bajas comisiones y un nivel de anonimato que protege la privacidad del usuario [8].

Más allá de las criptomonedas, esta tecnología se destaca por su naturaleza descentralizada, donde cada individuo en la red tiene acceso a una copia del registro completo de transacciones, lo cual garantiza una transparencia sin precedentes. Siendo así que la Blockchain cuenta con la capacidad para ofrecer la trazabilidad completa en las cadenas de suministro. Cada producto puede ser rastreado desde su origen hasta el consumidor final, asegurando la autenticidad y facilitando la detección de cualquier problema en el proceso [13].

La seguridad es otro de los pilares fundamentales de la Blockchain, ya que la inmutabilidad del registro asegura que una vez la información ha sido añadida a la cadena, esta no podrá ser alterada, reforzando así la confianza en el sistema.

Desde el punto de vista operativo, la Blockchain ofrece eficiencias significativas al eliminar los intermediarios, reduciendo tanto los tiempos de procesamiento como los costos asociados, a parte de minimizar las posibilidades de error. Este aspecto es crucial en sectores como el financiero, donde los contratos inteligentes automatizan y aseguran la ejecución de acuerdos sin necesidad de intermediación.

Tipos de Blockchain

Existen diferentes tipos de redes, cada una diseña para satisfacer unas necesidades en cuanto a la privacidad, gobernanza y accesibilidad. Se pueden clasificar en cuatro categorías principales [13]:

- **Blockchains públicas:** Son completamente abiertas y cualquier persona puede unirse. Bitcoin y Ethereum son buenos ejemplos de Blockchains públicas, donde las transacciones y los datos son visibles para todos, manteniendo al mismo tiempo el anonimato de los usuarios. Han pavimentando el camino para un ecosistema de aplicaciones descentralizadas (dApps) y finanzas descentralizadas (DeFi).

- **Blockchains semiprivadas:** Son operadas por una única entidad con la posibilidad de restringir el acceso, ofreciendo un equilibrio entre el control y la descentralización. A diferencia de las redes privadas, las semiprivadas pueden permitir la participación de partes externas bajo ciertas condiciones, manteniendo un nivel significativo de control sobre la red. Este tipo de redes las ofrecen empresas como IBM (Hyperledger Fabric) utilizada en sectores como la salud y la financiación, permite a las organizaciones configurar redes donde los datos se comparten solo con los actores autorizados, mejorando la eficiencia y la seguridad. Ofreciendo opciones personalizables para empresas, equilibrando la privacidad con la innovación [7].

- **Blockchains privadas:** Son operadas por una única organización, permiten un control total sobre quién puede participar en la red. Estas redes limitan el principio de la descentralización pero ofrecen una solución eficaz para entornos empresariales que necesitan privacidad y eficiencia en procesos internos. Por ejemplo, para este proyecto se ha utilizado la herramienta Ganache, la cual simula una red privada, siendo de gran utilidad en la fase de desarrollo y pruebas.

- **Consorcio:** Representan un equilibrio entre los modelos públicos y privados, siendo operadas por un grupo de organizaciones en lugar de una única entidad. Esta posibilidad permite compartir la responsabilidad del mantenimiento de la red entre varios participantes, lo que las hace adecuadas para colaboraciones interempresariales. Un ejemplo real sería R3 Corda, que facilita transacciones eficientes y seguras entre instituciones financieras, reduciendo costos y tiempos de procesamiento [2].

3.2. Ethereum

3.3. Contratos inteligentes

La idea fue conceptualizada por primera vez por Nick Szabo en 1993, visionando una nueva forma de establecer acuerdos digitales. Sin embargo, la falta de una plataforma adecuada mantuvo esta idea en teoría hasta la llegada de la Blockchain con Bitcoin en 2009, y mas notablemente con Ethereum en 2014, que los contratos inteligentes se materializaron prácticamente gracias a la infraestructura que esta tecnología proporciona [4].

Un contrato inteligente es un código que ejecuta automáticamente los términos de un acuerdo entre partes. Los códigos, almacenados en la Block-Chain, son ejecutados automáticamente, cuando se cumplen unas condiciones predefinidas, haciendo cumplir un acuerdo entre dos partes no confiables sin la necesidad de un tercero de confianza. Los contratos inteligentes utilizan la tecnología Blockchain para almacenar reglas, ejecutar automáticamente acciones cuando se cumplen esas reglas y almacenar los resultados en la blockchain. Debido a su naturaleza inmutable y distribuida, ofrecen un nivel de seguridad y confianza superior al de los sistemas tradicionales. Así mismo, al eliminar los intermediarios, ofrecen una reducción de costos y una mayor rapidez en la ejecución de acuerdos.

Sin embargo, se enfrentan a desafíos en cuanto a cuestiones legales, la necesidad de recursos externos a la cadena de bloques, su naturaleza inmutable que dificulta la corrección de errores, problemas de escalabilidad y limitaciones del mecanismo de consenso. Las soluciones de Capa 2, como la Lightning Network y Ethereum Plasma, se diseñaron para abordar los desafíos de escalabilidad y eficiencia de las blockchain de Capa 1. Operan sobre la cadena principal para permitir transacciones más rápidas y con menores costos, manteniendo la seguridad y descentralización [11].

3.4. Tokenización

La tokenización [1] es el proceso de convertir la información delicada o activos del mundo real en representaciones digitales denominadas "tokens", dentro del ecosistema Blockchain. Este procedimiento juega un papel crucial en la protección de datos confidenciales al reemplazar la información original con un token único, el cual no tiene valor fuera de su contexto específico de uso.

La información sensible se almacena en la "bóveda de tokenización-[14] una infraestructura de almacenamiento segura donde los datos originales se cifran y aíslan. El acceso a la bóveda es solo posible a través de rigurosos controles de seguridad y claves de descifrado especificas. A diferencia de los métodos de cifrado que utilizan un algoritmo matemático para transformar datos en un formato ilegible que puede ser revertido usando un clave concreta, la tokenizción no mantiene una relación algorítmica con los datos originales. En consecuencia, los tokens generados no pueden ser revertidos sin un acceso autorizado a la bóveda de tokenización, lo que proporciona una capa adicional de seguridad. Por lo tanto, mientras los datos originales se almacenan en una bóveda de tokens segura, los tokens se distribuyen en sistemas internos para su utilización diaria.

Un elemento importante de los tokens es que, fuera de la relación financiera específica para la que fueron creados, carecen totalmente de valor. Ya que una función de los mismos es representar un valor específico en una relación determinada. Esta característica los distingue de las criptomonedas y otros activos digitales que pueden tener un valor en el mercado abierto. En el ámbito de los pagos y transacciones, los tokens permiten a las organizaciones procesar transacciones y almacenar información de clientes sin exponer los datos críticos a riesgos de seguridad. La generación de un token se realiza mediante contratos inteligentes en la Blockchain, que definen las reglas y la lógica para su emisión, transferencia y anulación. Los contratos inteligentes aseguran que el token sea único y esté vinculado de manera inmutable a los datos o activos correspondientes en la bóveda.

En el ecosistema blockchain existen diversos tipos de tokens diseñados para propósitos específicos [10]:

- **Tokens de Seguridad:** Representa inversiones digitales en activos reales como acciones o bonos, respaldado por activos tangibles y regulado por entidades gubernamentales.

- **Tokens de Gobernanza:** Permiten a los poseedores participar en la toma de decisiones dentro de una plataforma o protocolo, votando en cambios o propuestas.

- **Tokens de Utilidad:** Proporciona acceso a productos o servicios dentro de una plataforma blockchain, sin ser considerado un valor financiero.

- **Tokens Comunitarios:** Recompensan la participación en una comunidad, ofreciendo beneficios como acceso exclusivo o descuentos a los miembros activos.
- **Tokens Vinculados a valores:** Son digitales pero están respaldados por activos físicos como metales preciosos, permitiendo a los inversores negociar activos reales de manera digital.

Todos los tipos de tokens existentes se pueden clasificar en dos grandes grupos, los tokens fungibles y los tokens no fungibles.

Tokens fungibles

La definición de fungibilidad es esencial para entender los aspectos fundamentales de un token fungible. Tomando como referencia una definición ofrecida por el Tesoro Público del Gobierno de España, la fungibilidad se describe como la "Propiedad de un conjunto de valores que los hace plenamente equivalentes entre sí a efectos legales-[5]. La fungibilidad es un concepto que nos rodea en la vida cotidiana, siendo el dinero uno de los mejores ejemplos. Cuando se intercambia un billete de cinco euros por otro billete de cinco euros, se entiende que ambos tienen el mismo valor y son aceptados de la misma manera.

Este principio se puede extrapolar al mundo digital y a la Blockchain. Los token fungibles actúan de forma similar al dinero físico, siendo indistinguibles y equivalentes entre unidades del mismo tipo. El ejemplo más conocido es Bitcoin, convirtiéndolo en una herramienta poderosa para las transacciones digitales.

Tokens no fungibles (NFT)

Los NFT han emergido como una innovación disruptiva en el ámbito del comercio electrónico, especialmente en el mundo del arte digital. A diferencia de los tokens fungibles, cada NFT es una certificación criptográfica que contiene información y códigos de identificación únicos que los hacen irremplazables e intercambiables. Esta característica los hace particularmente adecuados para representar activos digitales únicos y derechos de propiedad en el mundo digital. En el contexto de los contratos laborales, los NFT pueden ser utilizados para tokenizar y asegurar la autenticidad de contratos individuales, garantizando que los términos acordados sean únicos y vinculados inequívocamente a las partes involucradas. [15]

ERC-721

ERC-721 es un estándar propuesto por el desarrollador Dieter Shirley a finales de 2017 que introdujo el concepto de tokens no fungibles en la red Ethereum. [6] Abriendo las puertas a una nueva dimensión de activos digitales únicos, a diferencia de los tokens fungibles basados en el estándar ERC-20.

La creación del estándar ERC-721 fue motivada por la creciente demanda de tokens digitales que pudieran representar de manera única activos individuales. La singularidad de los tokens ERC-721 les dota de gran utilidad en aplicaciones donde el ámbito de autenticidad y la propiedad exclusiva son cruciales. Su uso mas popular se enfoca en el mundo del arte, asegurando la autenticidad y unicidad de diferentes obras, aunque también ha tomado gran relevancia en el ámbito legal. Poniendo de ejemplo este proyecto, con el uso del estándar ERC-721 se puede tokenizar y autenticar contratos laborales, asegurando la transparencia y la inmutabilidad de los términos acordados, verificando el cumplimiento de los acuerdos.

Este estándar cuenta con una serie de propiedades técnicas que lo hacen versátil. Algunas de estas propiedades son la asignación de un nombre, la definición de un balance de tokens dentro de una dirección y la implementación de funciones que permiten la transferencia segura de la propiedad.

3.5. Nodo

3.6. Wallet

3.7. Referencias

3.8. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

Herramientas	App AngularJS	API REST	BD	Memoria
HTML5	X			
CSS3	X			
BOOTSTRAP	X			
JavaScript	X			
AngularJS	X			
Bower	X			
PHP		X		
Karma + Jasmine	X			
Slim framework		X		
Idiorm		X		
Composer		X		
JSON	X	X		
PhpStorm	X	X		
MySQL			X	
PhpMyAdmin			X	
Git + BitBucket	X	X	X	X
MikTEX				X
TEXMaker				X
Astah				X
Balsamiq Mockups	X			
VersionOne	X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

Primer item más información sobre el primer item.

Segundo item más información sobre el segundo item.

-

3.9. Tablas

Igualmente se pueden usar los comandos específicos de LATEXo bien usar alguno de los comandos de la plantilla.

Algunos conceptos teóricos de LATEX [1].

[1] Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz

4. Técnicas y herramientas

4.1. Herramientas

Se muestra a continuación las herramientas usadas a lo largo del desarrollo del proyecto.

React Native

La elección del framework adecuado juega un papel crucial en el éxito de un proyecto. Dicha elección se complica aún más cuando se carece de experiencia previa en el desarrollo de aplicaciones móviles. Dentro del gran abanico de posibilidades React Native y Flutter emergen como grandes líderes en el sector gracias a sus grandes comunidades y la abundancia de recursos en línea. Por otro lado se pueden descartar directamente frameworks como Swift debido a su exclusividad para aplicaciones para iOS debido a que el objetivo de este proyecto es desarrollar una aplicación para Android.

React Native se presenta como mi elección favorita, este es un framework de código abierto creado por Facebook orientado a la creación de aplicaciones nativas tanto en iOS como en Android. React Native esta basado en JavaScript y React, una biblioteca de JavaScript destinada a la creación de interfaces de usuario. Fue lanzado en 2015 con el propósito de superar las limitaciones del desarrollo de aplicaciones móviles basado solo en HTML5, en las que simplemente se adapta aplicaciones web a un entorno móvil. React Native utiliza componentes nativos en lugar de WebViews para la interfaz de usuario, esto conlleva a que las aplicaciones se sientan y actúen como en una aplicación nativa. Uno de los elementos mas importantes de este framework es el React Native Bridge", el cual facilita la comunicación entre el código JavaScript y los elementos nativos del dispositivos. El puente

maneja de forma paralela dos flujos de trabajo, uno ejecuta la lógica de la aplicación en JavaScript y otro gestiona las operaciones de la interfaz de usuario nativa. Esto permite que las aplicaciones en React Native accedan a las características del dispositivo como la cámara o la ubicación manteniendo una experiencia fluida para el usuario.

Mi preferencia de este framework sobre el resto radica en mi familiaridad previa con la programación web y JavaScript, lo cual reducirá la curva de aprendizaje en la transición hacia React Native en contraste con otros frameworks que podrían requerir el aprendizaje de nuevos lenguajes de programación o paradigmas de programación. Por otro lado, uno de los grandes motivos de este framework es su gran popularidad, el cual se traduce en un gran riqueza de recursos disponibles, como bibliotecas, videotutoriales o foros, crucial a la hora de enfrentar los desafíos que puedan surgir. Finalmente aunque para este proyecto no sea un requerimiento hay que tener en cuenta la gran versatilidad de React Native, el cual permite la creación de aplicaciones nativas tanto en Android como en iOS a partir de un único código base. Optimizando así el proceso de desarrollo permitiendo en un futuro poder dar cobertura a un mercado más amplio sin esfuerzos duplicados.

Como había nombrado anteriormente Flutter es otro framework que destaca sobre el resto y ofrece ciertas ventajas notables sobre Reac Native, como un rendimiento superior gracias al uso de su motor de renderizado propio y la gran capacidad de personalización de la interfaz de usuario. Aunque Flutter pudiera ser superior en algunos aspectos técnicos sigo decantándome por React Native debido a su gran comunidad y la familiaridad que tengo con las tecnologías web, priorizando así un aprendizaje más sencillo frente a posibles mejoras en el rendimiento. Por otro lado existen frameworks Ionic y Xamarin que los he descartado debido a sus limitaciones en términos de acceso a funciones nativas o una comunidad bastante inferior comparado con React Native o Flutter. Kotlin era un opción con gran potencial para el desarrollo nativo de aplicaciones Android pero la descarte rápidamente debido a que disponía de una curva de aprendizaje más pronunciada que su competencia y iba a representar un obstáculo significativo en cuanto al tiempo de desarrollo sin garantizar beneficios proporcionales a dicho esfuerzo.

Respecto a Angular, si bien este framework ofrece un ecosistema robusto para el desarrollo de aplicaciones web dinámicas y complejas utilizando TypeScript, es importante mencionar que también es posible desarrollar aplicaciones móviles con Angular, ofreciendo una experiencia cercana a la nativa, aunque a través de un enfoque diferente al de las aplicaciones nativas

desarrolladas con React Native. Sin embargo, para el desarrollo específico de aplicaciones móviles nativas, React Native encaja mejor con los objetivos del proyecto. Angular no deja de ser una elección excelente, pero debido a la preferencia del proyecto de enfocarse en una aplicación móvil sin la necesidad de dar soporte de navegador, Angular puede no ser la opción más adecuada para el proyecto. Por lo tanto he descartado Angular buscando una solución más enfocada y optimizada para el desarrollo móvil, aprovechando las capacidades nativas de los dispositivos móviles.

Expo

Expo es un conjunto de herramientas, librerías y servicios para desarrollar aplicaciones nativas utilizando en Javascript y React Native. Proporciona un entorno de trabajo rico en funcionalidades que evita las complicaciones asociadas con la configuración nativa. Expo ha sido fundamental en el proyecto para desarrollar la interfaz de usuario permitiendo probar y visualizar cambios en tiempo real en diferentes plataformas. Una de las funcionalidades significativas de Expo ha sido su capacidad para facilitar la ejecución y testeo de la aplicación directamente en dispositivos móviles personales sin necesidad de emuladores. Esto se logra únicamente descargando la aplicación ExpoGo en el teléfono móvil y escaneando un código QR generado por el entorno de desarrollo Expo. Expo tiene un papel fundamental en las fases de prueba y depuración, permitiendo probar la aplicación en un entorno real y ajustar la interfaz la interfaz de usuario con un ciclo de retroalimentación casi instantáneo.

Solidity

Solidity es un lenguaje de programación orientado a objetos de alto nivel diseñado para escribir Smart Contracts que se ejecutan en la Ethereum Virtual Machine (EVM). Solidity consta de una sintaxis similar a lenguajes como JavaScript, C++ y Python, lo que facilita su aprendizaje. Es un lenguaje de tipado estático, lo que significa que el tipo de cada variable se define y no cambia durante al ejecución del contrato. Una de las potencialidades más destacadas de Solidity es su soporte para la herencia, una característica común en la programación orientada a objetos. Esto permite que los Smart Contracts hereden propiedades y comportamientos de otros contratos, beneficiándose de la reutilización de código y la organización de la lógica del mismo. En mi caso de gran utilidad para utilizar las funcionalidades del estándar ERC721 proporcionado por OpenZeppelin, que ofrece un conjunto de contratos inteligentes auditados y aprobados por su comunidad.

Remix

Remix es un entorno de desarrollo integrado (IDE) diseñado para el desarrollo de Smart Contracts escritos en Solidity. Proporciona una interfaz accesible y fácil de usar para escribir, compilar, probar y desplegar Smart Contracts directamente desde el navegador, sin la necesidad de instalar software adicional. Remix también ofrece funcionalidades avanzadas como la compilación en tiempo real, el despliegue de contratos en diversas redes de prueba (testnets) y la interacción con SmartContracts ya desplegados. Una de las características más útiles de este IDE es su análisis estático del código, pudiendo así identificar posibles errores de programación o vulnerabilidades de seguridad. Crucial para el desarrollo de Smart Contracts donde los errores pueden suponer consecuencias financieras significativas. Además, Remix se integra con herramientas y plugins adicionales, ofreciendo un entorna mas rico y extenso permitiendo conectarse con herramientas y servicios como MetaMask, Truffle y Ganache ampliamente utilizados en mi proyecto. Remix ha tenido una gran protagonismo en el desarrollo de mi proyecto permitiéndome iterar rápidamente a través de diferentes versiones de contratos inteligentes. Pudiendo ejecutar pruebas unitarias con diversas redes Ethereum como Rinkeby y Goerli, indispensable para validar la lógica del Smart Contract antes de su despliegue final. Dándome una gran capacidad de testeo en un entorno controlado pero realista el cual ha sido crucial para la corrección de errores y la optimización del uso de gas, asegurando así la eficiencia y seguridad de los contratos inteligentes desarrollados.

Truffle

Tras una fase inicial de desarrollo de los Smart Contracts usando Remix, la transición al uso de Truffle ha marcado un punto de inflexión en la complejidad de mi proyecto. Truffle consiste en una suite de desarrollo avanzada para Ethereum, ofreciendo un conjunto de herramientas diseñadas para facilitar el desarrollo, gestión y despliegue de Smart Contracts. Truffle proporciona un entorno de desarrollo estructurado generando una jerarquía de carpetas para favorecer la implementación de proyectos blockchain complejos. Ahorra mucho tiempo con su sistema de migraciones y scripts de despliegue el cual automatiza y simplifica el proceso de lanzamiento de contratos. Aunque Remix es una excelente herramienta, Truffle eleva la posibilidad de hacer pruebas unitarias a otro nivel con un marco de prueba mucho mas sofisticado, permitiendo ejecutar test en Solidity o JavaScript. Finalmente, uno de las mayores ventajas de usar Truffle en el desarrollo

de aplicaciones descentralizadas es en cómo simplifica el proceso de unir el trabajo que se realiza en el backend con el frontend. Cuenta con herramientas que ayudan a conectar Smart Contracts con aplicaciones móviles haciendo que la conexión sea mas directa y menos propensa a errores.

OpenZeppelin

OpenZeppelin es una biblioteca para el desarrollo seguro de Smart Contracts en Ethereum. Ofrece implementaciones auditadas y probadas para minimizar riesgos de seguridad. Su uso ha sido crucial para el desarrollo de mi proyecto, usando el estándar ERC721 para la representación de los contratos como tokens no fungibles (NFTs) asegurando que la aplicación cumpla con los estándares de seguridad.

Ganache

Ganache funciona como un nodo de Ethereum personal, permitiendo a los desarrolladores simular un entorno de blockchain que opera localmente ofreciendo un espacio seguro y controlado para experimentar sin ningún costo real ni tiempos de espera con las redes públicas de Ethereum. Proporciona diez cuentas cargadas con 1000 ETH cada una y con su clave pública y privada correspondiente. Ganache se puede usar tanto en la línea de comandos como en en la aplicación de escritorio y proporciona una vista de las transacciones, bloques y estado de la red, mostrando el feedback en tiempo real, vital para realizar una iteración rápida. Ganache se integra perfectamente con Truffle, siendo Ganache el entorno de desarrollo local predeterminado de Truffle.

MetaMask

MetaMask es una extensión de navegador y una aplicación móvil que permite a los usuarios interactuar con la blockchain de manera segura y sencilla. Actúa como puente entre los navegadores web y la blockchain. Su funcionamiento es análogo a una cartera digital, permitiendo a los usuarios almacenar sus cuentas de Ethereum. Al conectarse a dApps, los usuarios pueden usar sus cuentas de MetaMask para autenticarse eliminando la necesidad de copiar claves privadas manualmente. Por otro lado, aplica una capa adicional de seguridad ya que esta herramienta encripta la información del usuario y almacena las claves privadas directamente en el dispositivo del usuario. Esto asegura que solo el usuario tenga acceso a sus fondos y datos. La integración de MetaMask en mi proyecto no solo mejora la

experiencia del usuario final, sino que también agiliza el desarrollo de mi proyecto al proporcionar una manera fácil y segura de acceder a los activos de los usuarios.

Node.js

Node.js es un entorno de ejecución en JavaScript, tradicionalmente un lenguaje de programación del lado del cliente, para desarrollar aplicaciones del lado del servidor. Es conocido por su capacidad para manejar operaciones asíncronas y por su escalabilidad siendo de gran popularidad en el desarrollo de aplicaciones web modernas. Una de las ventajas de de Node.js es su ecosistema de paquetes gestionados por npm (Node Package Manager), que proporciona acceso a miles de librerías. En el proyecto se ha usado de manera frecuente, más allá de ser un requisito para utilizar Truffle, a través del uso de npm install se ha usado "npm install"para descargar librerías y paquetes necesarios para el desarrollo del frontend y backend. Bien es así, que aunque React Native sea el framework principal del desarrollo del frontend, la gestión de sus dependencias, librerías adicionales se realiza mediante npm, el cual opera sobre Node.js. A su vez, por ejemplo el uso de la biblioteca Web3.js, una de las mas importantes del proyecto ya que es fundamental para interactuar con la blockchain, se ha instalado y gestionado usando npm. Por tanto, Node.js ha sido una pieza crucial en el desarrollo del proyecto, simplificando la configuración del proyecto.

Mendeley

5. Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros3, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

6. Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

7. Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

[1] ASOBANCARIA. Hablemos de blockchain: ¿qué es la tokenización y para qué sirve? https://www.sabermassermas.com/hablemos-de-blockchain-que-es-la-tokenizacion-y-para-que-sirve/, 2021.

[2] André Carneiro. ¿what is r3 corda and how it works | bbchain? https://www.bbchain.com.br/en/blockchain-blog/what-is-r3-corda-and-how-it-works, 2022.

[3] Michele D'Aliessi. How does the blockchain work? https://onezero.medium.com/how-does-the-blockchain-work-98c8cd01d2ae, 2016.

[4] Universidad de Alcalá. Historia de los smart contracts. https://masterethereum.com/historia-smart-contracts/, 2019.

[5] Gobierno de España. Fungibilidad. https://www.tesoro.es/fungibilidad/, 2021.

[6] Juan Fornell. ¿qué es un token erc 721? https://academy.bit2me.com/que-es-token-erc-721/, 2019.

[7] IBM. ¿qué es hyperledger fabric? https://www.ibm.com/es-es/topics/hyperledger#:~:text=Hyperledger%20Fabric%20es%20una%20plataforma,con%20permisos%22%20(conocidos)., 2021.

[8] Lisa Institute. Qué es bitcoin: origen, usos, ventajas y riesgos. https://www.lisainstitute.com/blogs/blog/que-es-bitcoin-origen-usos-ventajas-riesgos, 2021.

[9] KeepCoding. ¿qué es el doble gasto? https://keepcoding.io/blog/que-es-el-doble-gasto/#:~:text=El%20doble%20gasto%20en%20las,digitales%20que%20representan%20la%20moneda./, 2023.

[10] KeepCoding. ¿qué tipos de tokens existen? https://keepcoding.io/blog/que-tipos-de-tokens-existen/, 2023.

[11] Ghedira-Guegan Chirine Khan Shafaq Naheed, Loukil Faiza and Bani-Hani Anoud Benkhelifa Elhadj. Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-Peer Networking and Applications*, 14(5):2901–2925, 2021.

[12] John Lee Quigley and John Gilbert. What is proof-of-work (pow)? all you need to know. https://blockworks.co/news/what-is-proof-of-work, 2023.

[13] SAP. ¿qué es la tecnología de blockchain? https://www.sap.com/spain/products/artificial-intelligence/what-is-blockchain.html, 2023.

[14] SoluLab. Vaultless tokenization vs. vault tokenization. https://www.solulab.com/vaultless-tokenization-vs-vault-tokenization/, 2019.

[15] Hamed Taherdoost. Non-fungible tokens (nft): A systematic review. *MDPI*, 14(26):26–, 2022.

[16] Wikipedia. Distributed ledger technology (dlt). https://es.wikipedia.org/wiki/Distributed_Ledger_Technology_(DLT), 2022.