
Galería de NFT's en Web3

TITULACIÓN: Grado en Ingeniería Informática

AUTOR: Cristóbal José Jiménez Gómez

TUTORIZADO POR:
María Dolores Afonso Suárez

Fecha: 05/2023

Agradecimientos

A los gnomos de jardín.

Resumen

Desarrollo de una Aplicación Web en el marco de la Web3(blockchain), que gestiona una galería de Non Fungible Tokens (NFT's). Ofrece la gestión de contenidos a distintos tipos de usuarios cuyos niveles de interacción varían desde los usuarios sin registrar, a administradores. En esa escala de accesos los casos de uso establecerán el nivel de interacción de cada perfil. Las funcionalidades implementadas permitirán la gestión a distintos niveles de estos NFT's, desde la visualización hasta la gestión de cada cartera. Este último concepto es el definido para almacenarlos y, en caso de ser necesario comerciar (compraventa o intercambio) con ellos.

Palabras claves: Web3, Web3.0, Angular, Firebase, Vercel, Express

Abstract

Development of a Web Application within the framework of Web3 (blockchain), which manages a gallery of Non Fungible Tokens (NFT's). It offers content management to different types of users whose levels of Interaction range from unregistered users to administrators. At this access scale, the use cases will establish the level of interaction of each profile. The functionalities implemented will allow the management at different levels of these NFTs, from the visualization to the management of each briefcase. This last concept is the one defined to store them and, in case it is necessary to trade (purchase or exchange) with them.

Key Words: Web3, Web3.0, Angular, Firebase, Vercel, Express

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Estructura de la memoria	2
2. Competencias específicas	4
3. Objetivos iniciales y estado actual	6
4. Marco teórico	8
4.1. Historia de la Web hasta la Web 3.0	8
4.2. Concepto de Cryptomoneda	9
4.3. Concepto de NFT	11
4.4. Concepto de CryptoCartera	13
5. Estado del Arte	15
5.1. Industria de la Web 3.0	15
5.2. Páginas principales de la Web 3 en sus distintos ámbitos	16
5.3. Conclusión	21
6. Recursos y tecnologías	23
6.1. Angular	23
6.2. Express.JS	24
6.3. API's	25
6.3.1. Firebase	25
6.3.2. Alchemy	27
6.3.3. Moralis	28
6.4. IDE - Visual Studio Code	28
6.5. Trello	29
6.6. Figma	30
6.7. Validaciones	30
6.7.1. Lighthouse	31
6.7.2. SonarQube	31
6.8. Control de versiones	32

<i>ÍNDICE GENERAL</i>	4
6.9. Git	32
6.10. GitKraken	33
7. Metodología	34
7.1. Marco de Trabajo SCRUM	34
8. Análisis y Diseño	36
9. Desarrollo	37
10. Evaluación y resultados	38
11. Conclusiones y trabajo futuro	39

Índice de figuras

4.1. Pegatina del servidor del CERN con la WWW	8
4.2. Las redes más populares según Metamask	10
4.3. Muestra de algunos NFT's de la colección de Moonbirds	11
4.4. Muestra de algunos NFT's de la Sorare	12
4.5. Criptocartera Hardware	13
4.6. Criptocartera Online	13
5.1. Logo de Steemit	16
5.2. Logo de Coinbase	17
5.3. Logo de Sia	18
5.4. Logo de IPFS	19
5.5. Logo de Audius	20
5.6. logos de Brave	20
5.7. Navegadores más usados entre 2018 y 2023	21
5.8. Capitalización total del mercado de criptomonedas entre abri del 2017 y mayo de 2023	22
6.1. Logo de Angular	23
6.2. Logo de Angular Material	24
6.3. Muestra de tabla y paginator en Trabajo propio	24
6.4. Logo de ExpressJS	25
6.5. Logo de Firebase	25
6.6. Tabla de autenticación proporcionada por Firebase	26
6.7. Conjunto de colecciones y documentos proporcionada por Firebase	27
6.8. Logo de Alchemy	27
6.9. Logo de Moralis	28
6.10. Logo de Visual Studio Code	29
6.11. Logo de Trello	29
6.12. Logo de Figma	30
6.13. Logo de Lighthouse	31
6.14. Ciclo que propone SonarQube	31
6.15. Logo de Figma	32
6.16. Logo de Figma	33
7.1. Metodología SCRUM	34

Índice de tablas

2.1. Competencias cubiertas durante el desarrollo del proyecto.	5
--	---

Capítulo 1

Introducción

1.1. Motivación

La razón principal que se encontró para la elaboración de este trabajo ha sido el auge de la Blockchain durante estos últimos años. "Blockchain es un libro mayor compartido e inalterable que facilita el proceso de registro de transacciones y de seguimiento de activos en una red de negocios. Un activo puede ser tangible o intangible. Prácticamente cualquier cosa de valor, puede rastrearse y comercializarse en una red de blockchain, reduciendo así el riesgo y los costes para todos los involucrados"[1].

Esta tecnología ofrece la posibilidad de construir aplicaciones web que garanticen la privacidad y seguridad de los datos de un usuario. La Web 3 ofrece transparencia y trazabilidad a la hora de la interacción entre usuarios. Además, permite la creación o conversión de negocios ya afianzados a este entorno.

Además, se afianza una necesidad que encontraba yo, personalmente, a la hora de interactuar entre distintas carteras, que es la necesidad de tener todas en una misma ubicación.

Se debe constar que ya existen páginas afianzadas para el fin de mostrar y compra-venta de NFT's. Este trabajo busca el aprendizaje tanto del desarrollo de una aplicación web completa, como de aprender a realizar las llamadas y conexiones para recoger y mostrar datos de NFT's[2] y criptocarteras[3].

1.2. Objetivos

El objetivo principal del proyecto se centra en el desarrollo de un sitio web que utiliza el nuevo concepto de descentralización dentro del marco Web3. El resultado ofrece como funcionalidad principal la búsqueda de NFT's y el almacenamiento de distintas carteras de criptomonedas. Para ello se plantea un conjunto de objetivos generales:

1. Estudiar las características y evolución de la Web 3.0 y su relación con la Web3.
2. Realizar un estudio de las tecnologías y herramientas a emplear.
3. Establecer criterios para su selección (tanto las tecnologías como de las herramientas)
4. Seguir una metodología de desarrollo, las pruebas y documentación del proyecto.

1.3. Estructura de la memoria

La memoria está dividida en 10 capítulos diferentes. En ellos se recoge toda la información, desde el inicio del proyecto hasta las conclusiones. Así, la información recogida en cada capítulo se describe tal que:

El primer capítulo consta de la introducción al proyecto. En este se contextualiza el trabajo, se explica la motivación detrás de la elaboración del proyecto, se dan a conocer los objetivos a cumplir y se explica la estructura del documento.

El segundo capítulo lista y justifica las competencias de la titulación abarcadas durante el desarrollo del trabajo mediante las teras y actividades realizadas.

El tercer capítulo desarrolla el marco teórico, donde se explican y analizan algunos términos relevantes a este trabajo. Es el punto de partida, donde se realiza una investigación que contextualiza y justifica las acciones que se toman en el trabajo.

El cuarto capítulo abarca el estado del arte, que explica la actualidad del desarrollo Web junto con el desarrollo dentro de la Web3.0 y la popularidad entre los distintos frameworks¹

El quinto capítulo contiene los recursos y las tecnologías utilizadas para el desarrollo del trabajo. En este se definen y justifican.

El sexto capítulo se explica la metodología llevada a cabo para la realización del trabajo, se dan a conocer las diferentes fases y tareas que han existido durante su desarrollo.

El séptimo capítulo se incluye un análisis de los requisitos y el diseño de la página que se ha elaborado para el desarrollo.

El octavo capítulo se explica todo lo relacionado con el desarrollo de la página web.

El noveno capítulo se encuentra la evaluación de la calidad de la página, obtenidos a través de diversos programas o extensiones.

¹Un framework es una herramienta de programación que te permite desarrollar software proporcionando una estructura con componentes integrados que sirven de base para construir proyectos nuevos[4]

El décimo capítulo se incluye las conclusiones que se pueden obtener de este proyecto, tanto personales como objetivas, dando además una propuesta de mejora para un futuro desarrollo.

Finalmente se incluyen los anexos, los cuales también contienen las referencias bibliográficas

Capítulo 2

Competencias específicas

Las competencias aplicadas a este proyecto se pueden encontrar en la Tabla 2.1 y a continuación se listan: CI8, CI13, CI16, CI17, T12.

La competencia CI8 se justifica debido a la necesidad de analizar, diseñar y construir los distintos componentes de la página web, así como las estructuras de los distintos documentos de la base de datos.

La competencia CI13 se justifica por el uso de hacer llamadas a API's y la creación de un backend. Así como a la hora de desplegar la aplicación a internet.

La competencia CI16 se justifica debido a la metodología Scrum aplicada durante el transcurso del proyecto.

La competencia CI17 se justifica gracias al uso de los programas de caldiad utilizados para garantizar la accesibilidad y usabilidad de la página web.

La competencia T12 se justifica debido a que en eso ha consistido este proyecto:

Diseño de los distintos componentes y páginas de la aplicación.

Despliegue de la aplicación en sus diferentes ramas para poder ser **evaluadas** por distintos usuarios.

Selección de las herramientas como pueden ser el *framework* sobre el que se ha trabajado (Angular), o la página sobre la que se ha hecho el *despliegue*, siempre teniendo en *cuenta el coste y la calidad* del mismo.

Tabla 2.1: **Competencias** cubiertas durante el desarrollo del proyecto.

Código	Descripción
CI8	Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
CI13	Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los sistemas de información, incluidos los basados en web.
CI16	Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería del Software
CI17	Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.
T12	Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.

Fuente: Universidad de Las Palmas de Gran Canaria (2023)

Capítulo 3

Objetivos iniciales y estado actual

El principal objetivo de este Trabajo de Fin de Grado se centra en el desarrollo de un sitio que hace uso del nuevo concepto de descentralización dentro del marco de la web 3.0 (Ver la evolución de la web en la sección 4.1).

El resultado ofrece como funcionalidad principal la búsqueda de NFT's (Ver la descripción en la sección 4.3) y el almacenamiento de distintas carteras de criptomonedas (Ver la descripción en la sección 4.4).

Para ello se plantea un conjunto de objetivos generales:

1. Estudiar las características y evolución de la Web 3.0 y su relación con la Web3.
 - Como se puede ver en el Capítulo 4, se han estudiado tanto las características, como la evolución de las Web3, sus diversos usos en la actualidad.
2. Realizar un estudio de las tecnologías y herramientas a emplear.
 - Firebase es una de las principales herramientas de Google para el manejo de usuarios
 - Hay una gran cantidad de API's que nos ayudan a acceder a la Web3, entre ellas:
 - Coinbase: <https://docs.cloud.coinbase.com>
 - ThirdWeb: <https://thirdweb.com>
 - Moralis: <https://moralis.io>
 - Alchemy: <https://www.alchemy.com>
 - Para el backend se revisó el uso de Django y de Express.js:
 - Usé Django en las Prácticas Externas y pude ver que es de fácil uso y comprensión debido al uso de Python, pero eran necesarios demasiados pasos para el despliegue de la aplicación

- Con Express.js fue simple, ya que ejecutar un servidor que escuche un puerto en concreto, es la base del funcionamiento de este Framework
- Para desplegar la página se ha revisado el funcionamiento de Amazon Web Service, GitHub Pages y Vercel:
 - Amazon Web Services es multiusos, lo vi como "matar una mosca a cañonazos", algo demasiado grande para lo que iba a ser el proyecto
 - A la hora de desplegar un proyecto angular en GitHub Pages me resultó complicado, ya que se tenía que crear la build del proyecto, y luego asignar el fichero que se mostraba. Si en algún momento me olvidaba de hacer una build, no se actualizaba aquí, por lo que fue descartada
 - Vercel aportaba algo simple, ya que si era un proyecto Angular lo desplegaba directamente, y a la hora de desplegar el backend, hecho en Express.js resultó ser sencillo, con un fichero de configuración escrito gracias a la documentación de Vercel
- 3. Establecer criterios para su selección (tanto las tecnologías como de las herramientas)
Este apartado se comprobará mejor en el Capítulo 6
 - Debido al "poco tiempo de desarrollo" decidí escoger las herramientas según la complejidad de las mismas:
 - Para tener la información de los usuarios, las distintas colecciones a mostrar, etc. Se usó Firebase por ser de Google y de las más completas
 - Para la obtención de NFT's se hizo uso de la API Alchemy, que con la documentación que se encontraba en la web se hizo fácil de usar
 - Para la obtención del balance de las carteras a partir de su dirección pública se hizo uso de la API de Moralis, ya que pude encontrar diversas formas de hacerlo, pero esta fue la más efectiva
 - Para esta última parte se hizo un backend con Express.js, que nos permite hacer uso de la funcionalidad de JavaScript para hacerlo
 - Para el despliegue de la aplicación se hizo uso de Vercel, por su simplicidad a la hora de desplegar los proyectos y documentación
- 4. Seguir una metodología de desarrollo, las pruebas y documentación del proyecto.

Capítulo 4

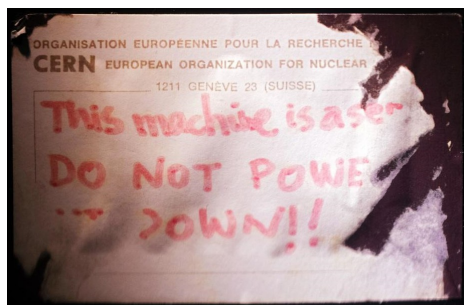
Marco teórico

En esta parte del informe se darían los conceptos generales para la comprensión del mismo. Se hará un paso por la historia de la Web desde sus inicios hasta como la conocemos ahora y se darán los conocimientos básicos para comprender el funcionamiento de las criptomonedas, los Token's no fungibles y las criptocarteras.

4.1. Historia de la Web hasta la Web 3.0

La **World Wide Web** fue creada en 1989 por el científico británico *Tim Berners-Lee* mientras trabajaba en el *Centro Europeo de Investigación Nuclear (CERN)*. La idea era crear una red de información que pudiera ser compartida entre científicos y académicos de todo el mundo. Para evitar un apagado accidental se escribió una nota en tinta roja (Figura 4.1) que ponía: "**This machine is a server. DO NOT POWER IT DOWN!!**"[5] (Esta máquina es un servidor. ¡¡NO LO APAGUEN!!)

Figura 4.1: Pegatina del servidor del CERN con la WWW



Fuente: Reddint oficial del CERN [<https://www.reddit.com/r/CERN/>]

In 1991, *Berners-Lee* creó una serie de tecnologías que permitían la conexión de documentos en un sistema hipertextual, utilizando el *protocolo HTTP* y la *codificación HTML*. Esto

permitió a los usuarios navegar por la red y acceder a documentos enlazados desde cualquier parte del mundo.

Con el tiempo, la Web se expandió y evolucionó, surgieron nuevas tecnologías como los motores de búsqueda, las redes sociales y las aplicaciones móviles, lo que llevó a la denominada **Web 2.0**.

La Web 2 se caracterizó por una mayor interactividad, el desarrollo de aplicaciones colaborativas y la creación de plataformas para la participación del usuario, como blogs, wikis y redes sociales. La Web2 también permitió la creación de empresas en línea y el desarrollo de nuevos modelos de negocio basados en la publicidad y los servicios en línea.[6]

La **Web 3**, también conocida como *Web descentralizada*, es la siguiente evolución de la Web. La Web 3.0 se centra en la descentralización de la web, lo que significa que los usuarios tienen mayor control sobre sus datos y pueden interactuar directamente entre sí sin la necesidad de intermediarios centralizados.

La tecnología clave detrás de la Web 3 es la cadena de bloques (blockchain) y otras tecnologías de registro distribuido (DLT), que permiten la creación de aplicaciones descentralizadas (dApps) y contratos inteligentes (smart contracts). Esto permite la creación de aplicaciones que no estén sujetas a la censura, la interferencia o la dependencia de un solo proveedor, lo que a su vez ofrece mayor privacidad y seguridad para los usuarios.

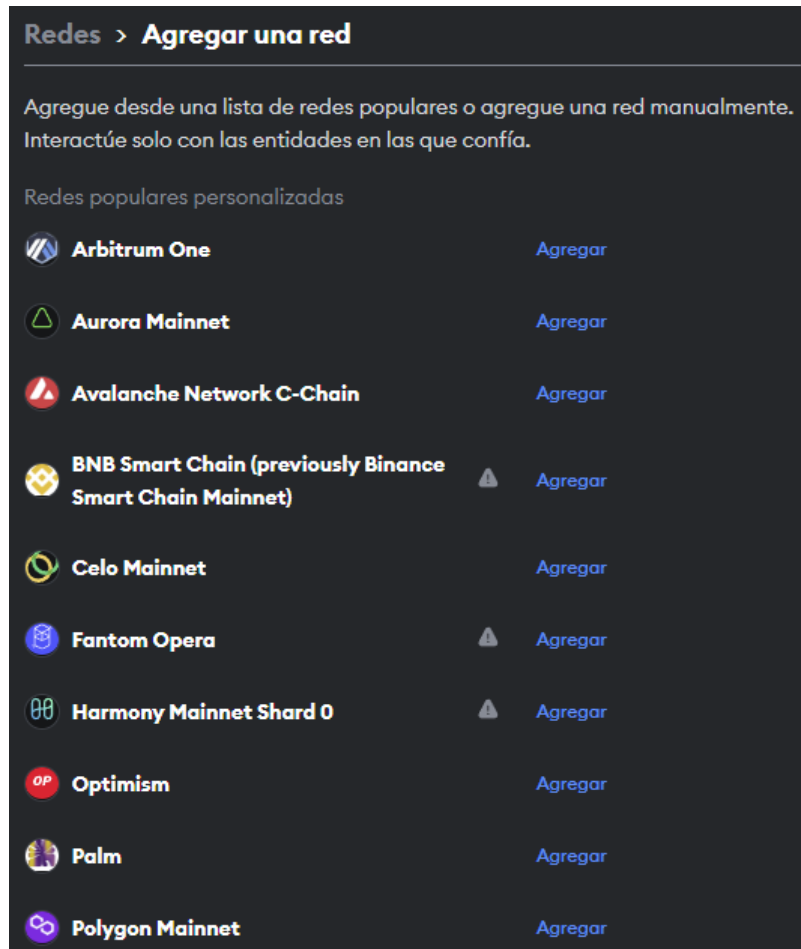
En resumen, la Web ha evolucionado desde su creación como WWW hasta la actualidad de la Web3. La Web 3 representa una evolución hacia un internet más descentralizado y democrático, donde los usuarios tienen un mayor control sobre sus datos y su experiencia en línea, y donde la confianza y la seguridad se pueden garantizar a través de la tecnología blockchain y otros mecanismos de confianza descentralizados.[7]

4.2. Concepto de Cryptomoneda

Las **criptomonedas** son monedas digitales que utilizan tecnología criptográfica para asegurar y verificar las transacciones y para controlar la creación de nuevas unidades de la moneda. Las criptomonedas funcionan en una red descentralizada, lo que significa que no están controladas por ningún gobierno, banco central o entidad financiera.

Cada criptomoneda tiene su propia **red de blockchain** (Figura 4.2), que es un registro público descentralizado que registra todas las transacciones de la moneda. La blockchain es una base de datos distribuida que contiene todas las transacciones realizadas con la moneda desde su creación, y se utiliza para verificar y validar las transacciones y para asegurar que no se puedan crear unidades adicionales de la moneda sin cumplir ciertas condiciones.

Figura 4.2: Las redes más populares según Metamask



Fuente: Cuenta personal de Metamask al querer agregar una nueva red a día 04/05/2023

Las **transacciones** en una criptomoneda se realizan de forma peer-to-peer (entre pares) y se validan a través de un proceso llamado **minería**. Esta es un proceso mediante el cual los usuarios de la red utilizan su poder de cómputo para resolver problemas matemáticos complejos y validar las transacciones en la blockchain. Los usuarios que realizan esta tarea reciben recompensas en forma de nuevas unidades de la criptomoneda.

Las criptomonedas se pueden comprar y vender en plataformas de intercambio de criptomonedas, y su valor depende de la oferta y la demanda del mercado.

Ofrecen varias ventajas, como la descentralización, la transparencia, la seguridad y la privacidad, pero también presentan algunos desafíos, como la volatilidad del valor y la falta de regulación.

En resumen, las criptomonedas son monedas digitales que utilizan tecnología criptográfica y una red descentralizada para asegurar y verificar las transacciones. Las criptomonedas se pueden comprar y vender en plataformas de intercambio, y su valor depende de la oferta

y la demanda del mercado. Las criptomonedas como todo en la vida, tiene sus ventajas y sus riesgos.

4.3. Concepto de NFT

Los **tokens no fungibles** (NFT, por sus siglas en inglés) son activos digitales únicos que se utilizan para representar elementos digitales como obras de arte, videos, música, juegos, entre otros. A diferencia de las criptomonedas tradicionales, los NFT no son intercambiables y cada uno es único e irrepetible.

Los NFT se basan en la **tecnología blockchain** y utilizan contratos inteligentes (**smart contracts**) para garantizar la propiedad y autenticidad del activo digital que representan. Los contratos inteligentes se utilizan para definir las condiciones y términos de la transacción, y garantizan que solo el propietario del NFT tenga derecho a la propiedad del elemento digital que representa.

Los NFT se han vuelto muy populares en el mundo del arte digital, ya que permiten a los artistas vender sus obras de arte como activos únicos [Por ejemplo, la figura 4.3] y garantizar su autenticidad y propiedad. También se están utilizando en otros sectores como los videojuegos, donde se pueden utilizar para representar objetos y personajes únicos [Por ejemplo, la figura 4.4].

Figura 4.3: Muestra de algunos NFT's de la colección de Moonbirds

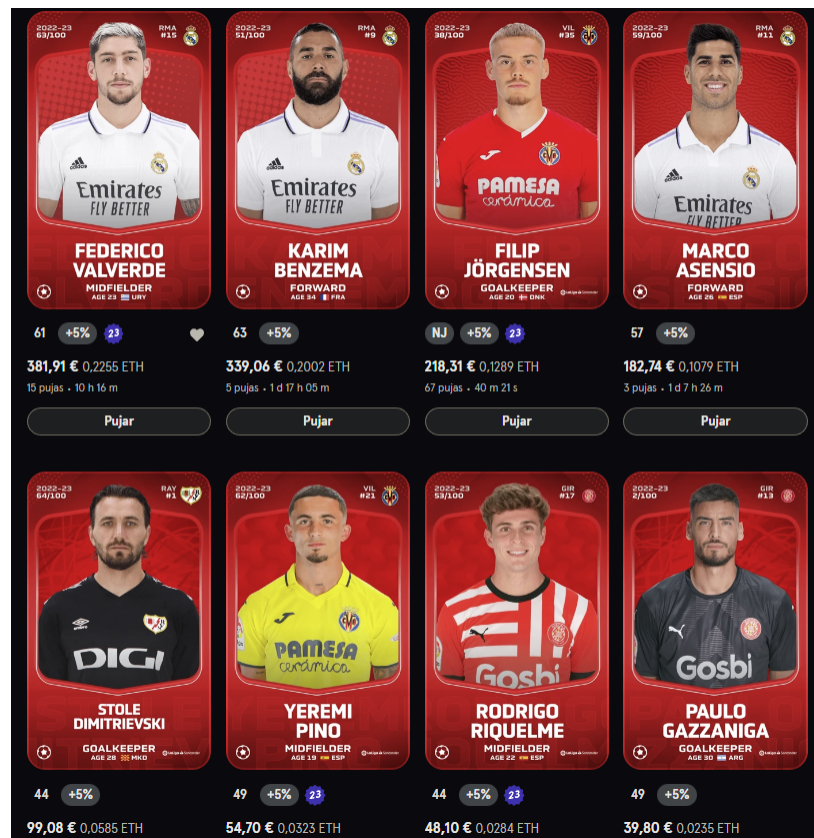


Fuente: Página de OpenSea [<https://opensea.io>]- Colección Moonbirds a día 03/04/2023

El valor de un NFT depende de la demanda del mercado y de la percepción del valor del activo digital que representa. Los NFT se pueden vender y comprar en plataformas especializadas y se utilizan principalmente como activos de inversión o de colección.

En resumen, los NFT son activos digitales únicos que se utilizan para representar elementos digitales como obras de arte, videos, música, juegos, entre otros, y se basan en la tecnología blockchain y contratos inteligentes para garantizar la propiedad y autenticidad del activo

Figura 4.4: Muestra de algunos NFT's de la Sorare



Fuente: Página de Sorare [<https://sorare.com/>]- Sección de mercado con filtros de jugador raro de la Liga Santander a día 03/04/2023

digital que representan. Los NFT se han vuelto muy populares en el mundo del arte digital y se están utilizando en otros sectores como los videojuegos[8].

4.4. Concepto de CryptoCartera

Una **criptocartera** es una herramienta que se utiliza para almacenar, enviar y recibir criptomonedas. Es similar a una billetera física, pero en lugar de contener billetes y monedas, una criptocartera contiene claves privadas y públicas que permiten acceder y administrar las criptomonedas.

Cada criptomoneda tiene su propia criptocartera, y hay varios tipos de criptocarteras disponibles, incluyendo carteras de hardware, software y en línea. Las **carteras de hardware** (Figura 4.5) son dispositivos físicos que se conectan a un ordenador y se utilizan para almacenar las claves privadas de manera segura. Las **carteras de software** se descargan en un ordenador o dispositivo móvil y se utilizan para almacenar las claves privadas en un archivo cifrado. Las **carteras en línea** (Figura 4.6) se almacenan en la nube y se pueden acceder desde cualquier dispositivo con una conexión a internet.

Figura 4.5: Criptocartera Hardware



Fuente: Página oficial de Ledger [<https://shop.ledger.com/products/ledger-nano-s-plus>]

Las criptocarteras son importantes porque las criptomonedas no se almacenan en un lugar

Figura 4.6: Criptocartera Online



Fuente: Logo oficial de Coinbase [<https://www.coinbase.com/es/>]

centralizado, como un banco, sino que se almacenan en la blockchain de la criptomoneda. Por lo tanto, es necesario utilizar una criptocartera para acceder y administrar las criptomonedas.

Las criptocarteras también permiten **enviar y recibir criptomonedas** de forma segura. Para enviar criptomonedas, se necesita la dirección pública de la criptocartera del destinatario. Para recibir criptomonedas, se proporciona la dirección pública de la criptocartera al remitente.

En resumen, una criptocartera es una herramienta utilizada para almacenar, enviar y recibir criptomonedas. Hay varios tipos de criptocarteras disponibles, incluyendo carteras de hardware, software y en línea. Las criptocarteras son importantes porque permiten acceder y administrar las criptomonedas, que se almacenan en la blockchain de la criptomoneda.

Capítulo 5

Estado del Arte

5.1. Industria de la Web 3.0

La industria de la web3.0 se compone de una variedad de proyectos y empresas que están trabajando en diferentes aspectos de esta tecnología. Algunos de estos proyectos incluyen:

Protocolos Blockchain

Son sistemas que permiten la creación y operación de redes descentralizadas y seguras. Estos protocolos permiten la creación de aplicaciones descentralizadas, contratos inteligentes y tokens criptográficos. Los protocolos blockchain están en constante evolución y crecimiento, y están impulsados por la comunidad y la colaboración. Algunos de los protocolos blockchain más populares son Bitcoin, Ethereum, Binance Smart Chain, Polkadot y Solana.

Criptomonedas

Son monedas digitales que se basan en la tecnología blockchain y se utilizan para realizar transacciones en línea de forma segura. Dentro de cada protocolo podemos encontrar distintas Criptomonedas, como pueden ser Bitcoin, Ethereum, Dólares Theter, Solana, Shiba Inu, etc.

Aplicaciones descentralizadas (dApps)

Este apartado se puede ver de forma más extendida dentro del subcapítulo 5.2.

Son aplicaciones web que se ejecutan en la blockchain y utilizan contratos inteligentes para permitir transacciones seguras y sin intermediarios. Las dApps pueden ser utilizadas para una amplia variedad de aplicaciones, desde finanzas descentralizadas (DeFi) hasta juegos y redes sociales. Algunos ejemplos de dApps populares incluyen:

- **Uniswap:** es una plataforma de intercambio descentralizada (DEX) que permite a los usuarios intercambiar criptomonedas sin la necesidad de intermediarios.

- **Decentraland**: es un mundo virtual descentralizado donde los usuarios pueden comprar, vender y construir propiedades virtuales utilizando criptomonedas.
- **Brave**: es un navegador web descentralizado que permite a los usuarios controlar su privacidad y monetizar su atención en línea.
- **OpenSea**: es un mercado de intercambio descentralizado para tokens criptográficos no fungibles (NFT) que permite a los usuarios comprar y vender obras de arte digitales, coleccionables y otros activos digitales únicos.

Conclusión

La industria de la web3.0 está en constante evolución y crecimiento, y está impulsada por la comunidad y la colaboración. Los proyectos y empresas en esta industria están trabajando juntos para construir una internet más segura, justa y descentralizada.

5.2. Páginas principales de la Web 3 en sus distintos ámbitos

A continuación veremos las **dApps** más conocidas a día de hoy en los distintos ámbitos principales[[9]]: Redes sociales, Servicios de intercambio monetario, servicios de almacenamiento, Servicios de *streaming* de vídeo y música

Redes Sociales Web3

Steemit[<https://steemit.com>]: Se ejecuta completamente en la blockchain Steem. Se describe mejor como una plataforma de recompensa descentralizada que ayuda a los contribuyentes a monetizar su contenido. Es una alternativa a Reddit[<https://reddit.com>]. Beneficios de

Figura 5.1: Logo de Steemit



Fuente: Steemit[<https://steemit.com>]

las redes sociales descentralizadas. Ninguna autoridad central que capture datos y los use. Empodera a los usuarios al recompensarlos con algún tipo de activo. Mejora en las redes sociales de la Web 2.0 en casi todos los sentidos. Protege la privacidad de los usuarios. Los usuarios deciden qué quieren compartir y cuándo. Las grandes corporaciones y organizaciones pierden poder para influir en las grandes corporaciones.

Servicios de intercambio descentralizados

Cualquiera de estas aplicaciones podrían asociarse a lo que podría ser un banco, donde podemos introducir dinero en distintas divisas, realizar transacciones, y el precio de las distintas divisas.

IDEX: Es un servicio de intercambio descentralizado popular para el comercio de tokens ERC-20. Proporciona una buena interfaz para los usuarios, y, cualquier persona con una cartera de ethereum puede comenzar a operar en la plataforma. Para hacer el mejor uso de IDEX o de cualquier intercambio descentralizado basado en ethereum, debes utilizar MetaMask.

Coinbase: Es una plataforma de intercambio de criptomonedas en línea que permite a los usuarios comprar, vender y almacenar una variedad de criptomonedas, como Bitcoin, Ethereum, Litecoin, entre otras. Fue fundada en 2012 y se ha convertido en una de las plataformas más populares y utilizadas en el mundo de las criptomonedas. Además de la plataforma de

Figura 5.2: Logo de Coinbase



Fuente: Coinbase[<https://www.coinbase.com>]

intercambio, Coinbase también ofrece una billetera digital integrada para almacenar las criptomonedas y una API para desarrolladores que deseen integrar las funcionalidades de la plataforma en sus propias aplicaciones.

Una de las principales características de Coinbase es su interfaz de usuario intuitiva y amigable, lo que la hace atractiva tanto para principiantes como para usuarios avanzados. También se ha destacado por su enfoque en la seguridad y la protección de los activos de los usuarios, utilizando medidas de seguridad avanzadas como la autenticación de dos factores y la custodia de criptomonedas.

Hacer intercambios de criptomonedas de forma descentralizada tiene ciertos beneficios:

- Transacciones más baratas.
- Transacciones más rápidas.
- Difícil de piratear debido a la naturaleza descentralizada.
- Funciona bien con carteras de hardware.
- Los usuarios controlan sus propios fondos.

Sin embargo, los intercambios descentralizados no están libres de negativos. Pueden ser difíciles de usar y comprender para los usuarios. Además, la generación actual de intercambios

descentralizados también sufre la falta de características y funcionalidades. Sin embargo, con el tiempo, veremos la red descentralizada más avanzada a la par con las contrapartes centralizadas.

Servicios de Almacenamiento descentralizados

Las siguientes herramientas podrían ser comparadas en la Web2 con Dropbox, Google Drive o Microsoft OneDrive.

Storj[<https://www.storj.io>]: Storj es una de las principales soluciones de almacenamiento descentralizado. También es uno de los más antiguos. Con Storj, cualquiera puede almacenar datos. También es de código abierto y fácil de usar. Cualquiera puede comenzar a utilizarlo con solo 1-Clic en Inicio. El modelo de pago se crea alrededor de los usuarios, ya que pueden pagar según lo utilicen. El token Storj se utiliza para alimentar la plataforma Storj.

Sia[<https://sia.tech>]: Proporciona almacenamiento de forma descentralizada y también se considera la mayor competencia de Storj. Sia divide el archivo en treinta segmentos y luego lo distribuye. También encripta el archivo mientras se transfiere. **IPFS (InterPlanetary**

Figura 5.3: Logo de Sia



Fuente: Steemit

File System)[<https://ipfs.tech>]: Es una tecnología de almacenamiento distribuido que utiliza la red blockchain para almacenar archivos de manera descentralizada. IPFS permite a los usuarios acceder a los archivos de forma más rápida y segura, ya que los archivos se almacenan en múltiples nodos en la red. Beneficios de las soluciones de almacenamiento descentralizado Funciona bien en diferentes plataformas o incluso en soluciones de blockchain. Protege los datos que se transfieren con cifrado fuerte. Ninguna entidad centralizada, significa que nadie puede usar los datos. Es barato y funciona bien con tecnologías de próxima generación como **IoT (Internet de las cosas)**.

Figura 5.4: Logo de IPFS



Fuente: wikipedia [https://es.wikipedia.org/wiki/Sistema_de_archivos_interplanetario]

***Streaming* de Vídeo y Música**

Aquí podríamos comparar con aplicaciones como Twitch[<https://www.twitch.tv>], YouTube[<https://www.youtube.com>] o U-beat[<https://ubeat.tv>], en el apartado de *streaming* de vídeo, y aplicaciones web como YouTube Music[<https://music.youtube.com>], Spotify[<https://open.spotify.com/>], o SoundCloud[<https://soundcloud.com>], para el *streaming* de música.

LivePeer[<https://livepeer.org/es>]: Proporciona un servicio de transmisión, es de código abierto y apunta a construir una stack de streaming para la Web 3.0.

LBRY[<https://lbry.com>]: Biblioteca digital descentralizada que alberga diferentes formas de contenido. Como usuario, puedes leer, ver y jugar en la plataforma. Esto significa que admite libros, música y videos. Es uno de los proyectos Web 3.0 más antiguos.

UjoMusic[<https://www.mesh.xyz>]: Plataforma de música donde los creadores pueden cargar su música y distribuirla sin problemas de derechos de autor o royalty. Las criptomoneda y los contratos inteligentes lo potencian.

Audius[<https://audius.co>]: Es una plataforma de música descentralizada que permite a los artistas compartir su música y monetizarla. Los usuarios pueden escuchar música de forma gratuita. Esta aplicación es la que pretende reemplazar a Spotify dentro del mundo descentralizado.

Los creadores de contenido pueden trabajar en un entorno transparente. todos tienen las mismas oportunidades de promover su trabajo. Los problemas de derechos de autor serán insignificantes gracias a los contratos inteligentes y no hay autoridad central, por lo tanto no será una política absurda para los streamers y los creadores de contenido.

Figura 5.5: Logo de Audius



Fuente: Ecosystem.ipfs [<https://ecosystem.ipfs.tech/project/audius/>]

Navegadores descentralizados

Teniendo en cuenta los navegadores más utilizados en estos últimos 5 años [Figura 5.7], podemos ver que las comparaciones son abrumadoras, teniendo Google Chrome casi un 63 % del mercado, siguiéndole Safari, Mozilla Firefox y Samsung Internet. Por lo que cualquiera de los siguientes navegadores no pueden competir con este, aunque tienen sus beneficios al estar implementando la tecnología blockchain.

Brave Browser[<https://brave.com/es/>]: Brave tiene que ver con la privacidad donde los usuarios no son el producto. El navegador viene preinstalado con el bloqueador de anuncios. También permitirá a los usuarios vender sus datos a cambio de la criptomoneda, BAT, la cual puede ser usada para diversos propósitos, como pueden ser hacer Staking, ser cambiadas por otra Criptomoneda, o venderla para conseguir divisa.

Figura 5.6: logos de Brave



Fuente: logos.Wine

Breaker Browser[<https://github.com/beakerbrowser>]: Es un navegador web de próxima generación basado en punto a punto (P2P)¹. Es un lugar donde cualquiera puede unirse, compartir y sobrecargar sus aplicaciones. Es una herramienta creativa que puede ser explorada por cualquier persona. Es un navegador web 3.0.

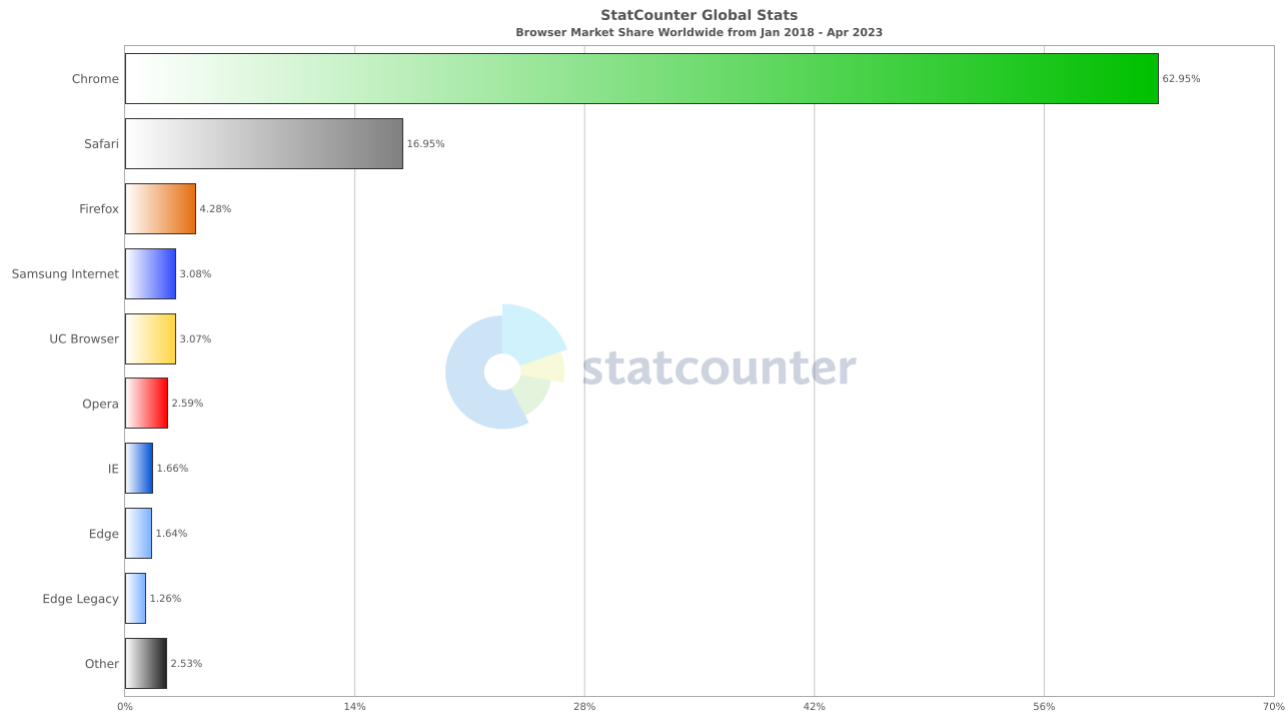
Beneficios de los navegadores descentralizados: Los usuarios pueden navegar de forma privada por internet.

¹red de computadoras en la que todos los aspectos (o la mayoría de ellos) funcionan con una serie de nodos que se comportan de la misma manera entre sí y sin clientes ni servidores fijos[10]

Ninguna o menor cantidad de lagunas de seguridad.

Los usuarios pueden vender sus datos a una organización y recibir pagos. Rápido y seguro.

Figura 5.7: Navegadores más usados entre 2018 y 2023



Fuente: StatCounter

[<https://gs.statcounter.com/browser-market-share#monthly-201801-202304-bar>]

5.3. Conclusión

La web3 se está desarrollando rápidamente y se espera que cambie la forma en que interactuamos en línea. Con la tecnología blockchain y los contratos inteligentes, la web3 permitirá aplicaciones descentralizadas, transacciones financieras sin intermediarios y la propiedad verdadera de los datos. Algunos de los protocolos blockchain más populares para la web3 incluyen Ethereum, Polkadot y Solana.

Las criptomonedas continúan siendo una fuerza importante en el mercado financiero global. Bitcoin sigue siendo la criptomoneda más grande y popular, pero hay muchas otras criptomonedas importantes, como Ethereum, Binance Coin y Cardano. La capitalización total del mercado de criptomonedas ha aumentado significativamente en los últimos años, llegando a más de 2 billones de euros [Figura 5.8] en abril de 2021.

Los NFTs se han convertido en un nuevo mercado emocionante en la web3. Las ventas de NFTs han aumentado significativamente desde 2020, con algunos NFTs vendiéndose por decenas de miles de dólares.

Figura 5.8: Capitalización total del mercado de criptomonedas entre abril del 2017 y mayo de 2023



Fuente: CoinMarketCap [<https://coinmarketcap.com/es/charts/>]

La regulación de las criptomonedas y los NFTs sigue siendo un tema importante. A medida que estos mercados crecen, los reguladores de todo el mundo están considerando cómo regularlos para proteger a los inversores y prevenir el fraude. En algunos países, como China, se han tomado medidas más drásticas para prohibir las criptomonedas y las transacciones de NFTs.

Capítulo 6

Recursos y tecnologías

En este capítulo se tratan los recursos y las tecnologías empleadas para el diseño y desarrollo de la aplicación Web.

6.1. Angular

Angular[<https://angular.io>] es un framework de desarrollo web para crear aplicaciones de una sola página (SPA) y aplicaciones web dinámicas. Fue desarrollado por Google y se basa en el lenguaje de programación TypeScript.

Figura 6.1: Logo de Angular



Fuente: Página oficial de Angular [<https://angular.io>]

Angular proporciona una estructura sólida y coherente para el desarrollo de aplicaciones web complejas, lo que permite a los desarrolladores construir aplicaciones de alta calidad y escalables de manera más rápida y eficiente. Algunas de las características clave de Angular incluyen:

Inyección de dependencias: Permite que los componentes se comuniquen entre sí de manera eficiente.

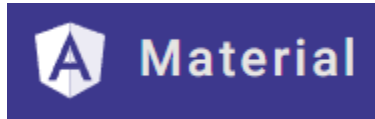
Binding de datos bidireccional: Facilita la actualización automática de la interfaz de usuario en tiempo real.

Directivas: Permite crear componentes personalizados y hacer uso de componentes predefinidos.

Servicios: Permite compartir datos y lógica de aplicación entre los componentes.

Angular también viene con una amplia variedad de herramientas y bibliotecas adicionales, como Angular CLI y Angular Material, que ayudan a los desarrolladores a crear aplicaciones web robustas y de alta calidad.

Figura 6.2: Logo de Angular Material



Fuente: Página oficial de Angular Material [<https://material.angular.io>]

Se ha hecho uso de un algunos componentes de Angular Material, como son:

Material Table [<https://v5.material.angular.io/components/table/overview>], junto con Paginator:

El **componente mat-table** proporciona una tabla de datos de estilo Material Design que se puede utilizar para mostrar filas de datos.

El **componente mat-paginator** Proporciona navegación para la información paginada, normalmente utilizada con una tabla.

Figura 6.3: Muestra de tabla y paginator en Trabajo propio

Fullname	Email	Create_date	Last_Login	Phone	Wallets	IsAdmin
Admin	admin@admin.com	Mon May 08 2023	Mon May 08 2023		1	true
Cristóbal José Jiménez Gómez	cristobal_jjg@hotmail.com	Sun May 07 2023	Sun May 07 2023	+34 615 31 62 91	1	true
user try 3	try_user@gmail.com	Invalid Date	Invalid Date		0	false
New Profile	tryuser@gmail.com	Sun May 07 2023	Sun May 07 2023		0	false
Items per page: 5 0 of 0 < > >>						

Fuente: Trabajo Propio en la zona de admin
[<https://tft-galeria-nft-web3.vercel.app/admin/users>]

6.2. Express.JS

Como dicen en su propia página web, *ExpressJS* es una opción rápida, sin opiniones y minimalista para Node.js.”

Es un popular marco de aplicación web para Node.js que simplifica el proceso de creación

de aplicaciones web y APIs (interfaces de programación de aplicaciones). Es una capa delgada sobre Node.js que proporciona una variedad de características y herramientas para crear aplicaciones web y APIs de manera rápida y eficiente.

Figura 6.4: Logo de ExpressJS



Fuente: Página oficial de ExpressJS [<http://expressjs.com>]

ExpressJS es un marco minimalista que proporciona una estructura básica para construir aplicaciones web, pero permite a los desarrolladores agregar funcionalidad adicional utilizando middleware y complementos. ExpressJS también proporciona un enrutamiento simple y flexible, lo que facilita la creación de rutas para diferentes páginas y recursos.

Además, ExpressJS es altamente personalizable y se integra fácilmente con otros módulos y herramientas de Node.js, lo que lo convierte en una opción popular para construir aplicaciones web y APIs.

6.3. API's

6.3.1. Firebase

Firebase es una plataforma de desarrollo de aplicaciones móviles y web que ofrece una variedad de herramientas y servicios en la nube para ayudar a los desarrolladores a crear, mejorar y escalar aplicaciones.

Figura 6.5: Logo de Firebase



Fuente: Página oficial de Firebase [<https://firebase.google.com/?hl=es>]

Firebase fue adquirido por Google en 2014 y desde entonces ha evolucionado para ofrecer una amplia gama de servicios, incluyendo:

- Autenticación de usuarios[Capítulo 6.3.1.1]: permite a los desarrolladores agregar fácilmente funciones de registro, inicio de sesión y autenticación a sus aplicaciones.
- Almacenamiento en la nube[Capítulo 6.3.1.2]: proporciona almacenamiento en la nube para archivos y datos de aplicaciones.

- Base de datos en tiempo real: una base de datos en tiempo real en la nube que permite a los desarrolladores crear aplicaciones en tiempo real con actualizaciones automáticas y en tiempo real.
- Alojamiento de aplicaciones web: permite a los desarrolladores alojar sus aplicaciones web en los servidores de Firebase.
- Notificaciones push: permite enviar notificaciones push a los usuarios de la aplicación.
- Analítica de aplicaciones: proporciona análisis de usuarios y datos de uso de la aplicación.

Para este proyecto no se ha hecho uso de todas las utilidades, solamente de:

6.3.1.1. Autenticación de usuarios

Se ha hecho uso de esta utilidad debido a que nos proporciona una forma sencilla de registrar usuarios, hacer que estos inicien sesión, y que puedan cerrar sesión de una forma simple. Además, y como veremos en las conclusiones y trabajo futuro [Capítulo ??], nos permite hacer uso de la autenticación de Google, lo que nos permite tener una mayor seguridad en el inicio de sesión de los usuarios.

Figura 6.6: Tabla de autenticación proporcionada por Firebase



Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
admin@admin.com	✉	8 may 2023	10 may 2023	TdSGbnUc1GZUiQx4Iuy0yFsJNjN2
tryuser@gmail.com	✉	7 may 2023	7 may 2023	B3US86o17PPfPRvtIABY8Rn4V0H2
crisobal_jjg@hotmail.com	✉	7 may 2023	9 may 2023	ZNDYxuyxSHWTTLevaUycyNH1yx...
try_user@gmail.com	✉	6 abr 2023	7 may 2023	dRitryCb0vgNuncMs2Efc8XfTTt2

Filas por página: 50 1 - 4 of 4

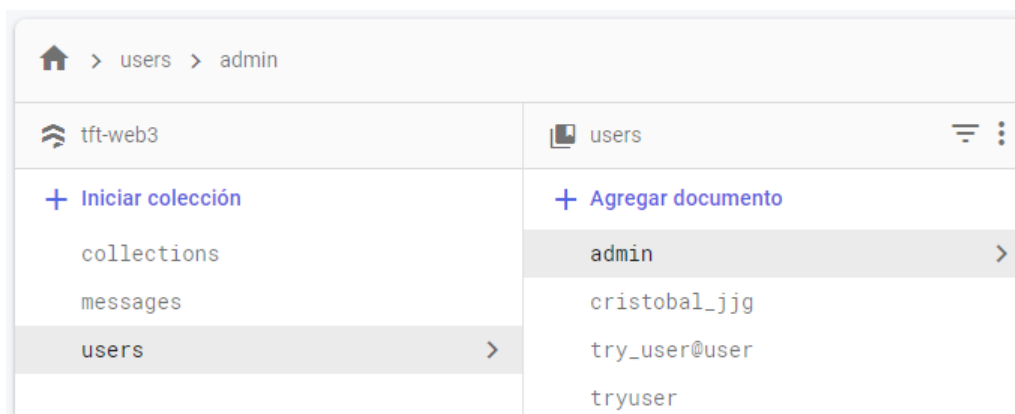
Fuente: Consola de Firebase, apartado de Autenticación

6.3.1.2. Almacenamiento en la nube

Se ha hecho uso de esta utilidad debido a que nos proporciona una base de datos en la nube, lo que nos permite tener una mayor seguridad en el almacenamiento de los datos de los usuarios, y de la aplicación en general. En este caso, guardamos 3 tipos de datos:

- Los **usuarios** registrados, con la información del correo, si es administrador, el nombre, el apellido, su número de móvil, su nombre de usuario, que está compuesto por el inicio del correo y su primer nombre, y las carteras que este ha agregado.
- Las **colecciones** que se mostrarán de NFT's, que contienen las direcciones de las colecciones para poder acceder a las mismas, la url de la colección, el nombre real y la foto de la marca.
- También se ha hecho un apartado para **mensajes**, el cual contiene mensajes que han enviado los usuarios con intenciones de mejora. Este documento contiene el correo del usuario que lo ha mandado, el mensaje, y un asunto.

Figura 6.7: Conjunto de colecciones y documentos proporcionada por Firebase



Fuente: Consola de Firebase, apartado de Firestore Database

6.3.2. Alchemy

Alchemy es una plataforma de desarrollo con soporte multicadena y con alcance global, pensada en facilitar el desarrollo de aplicaciones descentralizadas (DApps). Su principal objetivo es ofrecer todo lo que los desarrolladores necesitan para construir y hacer realidad la Web3.[[11]]

Figura 6.8: Logo de Alchemy



Fuente: Página oficial de Alchemy [<https://www.alchemy.com>]

Su relevancia en el sector le ha valido ser reconocida como el “AWS de la Web3”, manejando más de 10 millones de usuarios, movilizando más de 100 mil millones de dólares en activos digitales y con un concurrencia de más de 100 mil millones de requests, lo que le ha llevado a tener una valoración de mercado de más 10 mil millones de dólares.

La intención de la plataforma es permitir que las aplicaciones puedan evolucionar rápidamente para dar respuesta a las necesidades de los usuarios, sin que esto implique la puesta en marcha de tales mejoras. Así, los desarrolladores se pueden enfocar en lo realmente importante: diseñar y codificar estas nuevas soluciones, confiando en que la plataforma tendrá la flexibilidad necesaria para respaldar estos nuevos diseños y permitir que los usuarios puedan explorarlos.

6.3.3. Moralis

Moralis [<https://moralis.io>] es una plataforma de infraestructura blockchain de la web3 que proporciona herramientas y servicios para desarrolladores que construyen aplicaciones descentralizadas (dApps) en la cadena de bloques Ethereum. Ofrece una amplia gama de herramientas, incluyendo APIs y nodos de red, para que los desarrolladores puedan interactuar con Ethereum de manera eficiente y escalable.

Figura 6.9: Logo de Moralis



Fuente: Página oficial de Moralis [<https://moralis.io>]

Moralis también proporciona herramientas de desarrollo para que los desarrolladores puedan construir dApps en la cadena de bloques Ethereum de manera más fácil y rápida. La plataforma es utilizada por una gran cantidad de proyectos y empresas en el espacio de la web3, y es conocida por su facilidad de uso y eficiencia en la construcción de aplicaciones descentralizadas en la cadena de bloques Ethereum.

Entre estas empresas, podemos encontrar algunas antes nombradas, como son Metamask [<https://metamask.io>] o Polygon [<https://polygon.technology>]

6.4. IDE - Visual Studio Code

Visual Studio Code (también conocido como VS Code) [<https://code.visualstudio.com>] es un editor de código fuente desarrollado por Microsoft, que es compatible con múltiples lenguajes de programación. Está disponible de manera gratuita para Windows, Linux y macOS, y es muy popular entre los desarrolladores por su facilidad de uso, flexibilidad y personalización.

VS Code incluye características útiles como resaltado de sintaxis, autocompletado de código, depuración de código en vivo, integración con control de versiones y herramientas de construcción y despliegue de aplicaciones. Además, los usuarios pueden personalizar el editor con extensiones y temas para satisfacer sus necesidades específicas de programación.

Figura 6.10: Logo de Visual Studio Code



Fuente: Iconduck [<https://iconduck.com/icons/102490/file-type-vscode>]

VS Code es muy popular en el desarrollo web y en la construcción de aplicaciones para la web3 y las criptomonedas.

Se han utilizado algunas extensiones básicas para el desarrollo web como son:

- Angular Language Service [<https://angular.io/guide/language-service>]
- Auto Close Tag [<https://marketplace.visualstudio.com/items?itemName=formulahendry.auto-close-tag>]
- Auto Rename Tag [<https://marketplace.visualstudio.com/items?itemName=formulahendry.auto-rename-tag>]
- Console Ninja [<https://marketplace.visualstudio.com/items?itemName=WallabyJs.console-ninja>]
- IntelliCode [<https://marketplace.visualstudio.com/items?itemName=VisualStudioExptTeam.vscointellicode>]
- GitHub Copilot Nightly [<https://marketplace.visualstudio.com/items?itemName=GitHub.copilot-nightly>]
- Prettier - Code formatter [<https://marketplace.visualstudio.com/items?itemName=esbenp.prettier-vscode>]
- SCSS Formatter [<https://marketplace.visualstudio.com/items?itemName=sibiraj-s.vscode-scss-formatter>]
- VSCode-pdf [<https://marketplace.visualstudio.com/items?itemName=tomoki1207.pdf>]

6.5. Trello

Trello es una herramienta de gestión de proyectos basada en la nube, que permite a los usuarios organizar tareas y proyectos en tableros visuales y colaborativos. Fue lanzado en 2011 y se ha vuelto muy popular entre equipos de trabajo de diversas industrias y tamaños.

Figura 6.11: Logo de Trello



Fuente: Iconduck [<https://iconduck.com/icons/95002/trello>]

Trello utiliza un sistema de tableros que representan los diferentes proyectos o áreas de trabajo, y dentro de cada tablero, los usuarios pueden crear listas de tareas y tarjetas que representan cada tarea o actividad. Estas tarjetas pueden contener información como descripciones, listas de verificación, etiquetas, fechas de vencimiento, comentarios y archivos adjuntos.

Además, Trello permite la colaboración entre equipos de trabajo, ya que los miembros pueden comentar en las tarjetas, asignar tareas a otros miembros, establecer fechas de vencimiento y recibir notificaciones de actualizaciones. Trello también se integra con otras herramientas populares de productividad, como Slack, Google Drive y Jira, lo que lo hace aún más útil para equipos de trabajo que utilizan diferentes herramientas.

6.6. Figma

Figma es una herramienta de diseño gráfico en línea que se utiliza para crear interfaces de usuario, diseños de sitios web, aplicaciones móviles y otros proyectos digitales. Fue lanzado en 2016 y se ha vuelto muy popular en la industria del diseño debido a su capacidad para trabajar en tiempo real y permitir la colaboración en tiempo real entre los miembros del equipo. Figma nos permite crear diseños con herramientas como vectores, formas, capas,

Figura 6.12: Logo de Figma



Fuente: Iconduck [<https://iconduck.com/icons/39864/figma>]

texto y efectos, entre otras. También tiene características como prototipado y animación, lo que lo hace ideal para diseñar interfaces interactivas.

Además, Figma es una herramienta basada en la nube, lo que significa que todos los archivos se guardan en línea y se pueden acceder desde cualquier lugar con conexión a Internet. Esto permite la colaboración en tiempo real entre los miembros del equipo, ya que varios diseñadores pueden trabajar en un proyecto simultáneamente y realizar cambios que se actualizan en tiempo real.

6.7. Validaciones

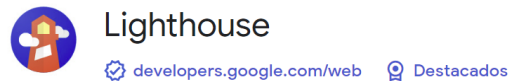
En este último curso se ha profundizado intensamente sobre la calidad del producto y se han usado algunas herramientas para comprobar objetivamente la calidad del mismo. En esta sección se tratarán las herramientas usadas para validar el proyecto de forma objetiva.

6.7.1. Lighthouse

Lighthouse es una herramienta automatizada de código abierto para mejorar el rendimiento, la calidad y la corrección de sus aplicaciones web.

Al auditar una página, Lighthouse ejecuta un aluvión de pruebas contra la página y luego genera un informe sobre qué tan bien lo hizo la página. Desde aquí puede utilizar las pruebas de falla como indicadores de lo que puede hacer para mejorar su aplicación.[12]

Figura 6.13: Logo de Lighthouse



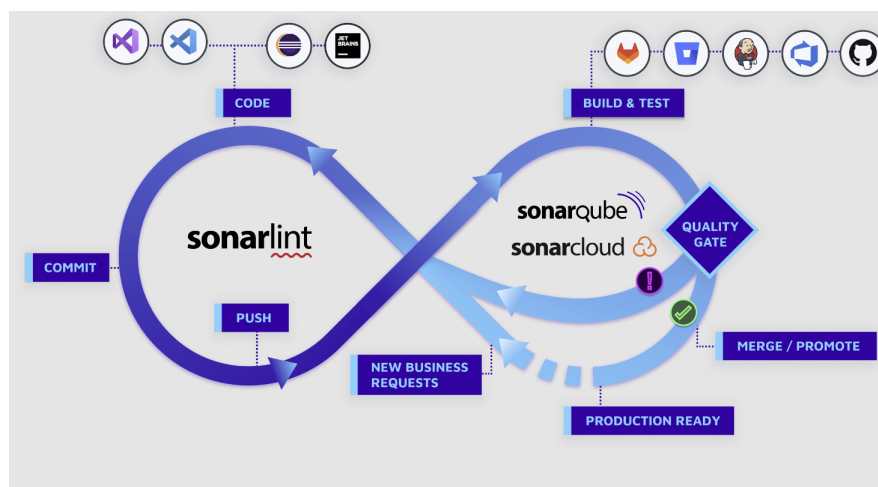
Fuente: Chrome Web Store [<https://chrome.google.com/webstore/detail/lighthouse/blipmdconlcpinefehnmjammfjpmpbjk?hl=es>]

Con esta herramienta se pretende obtener objetivamente una versión más accesible de la aplicación, que los colores tengan una coherencia, sean visibles de una forma correcta sin contrastes extraños o disonancias de tonos, así como una ayuda a la hora de controlar el SEO de la aplicación.

6.7.2. SonarQube

SonarQube es una herramienta de revisión de código automática y autogestionada que ayuda sistemáticamente a entregar código limpio. SonarQube se integra en su flujo de trabajo existente y detecta problemas en su código para ayudarlo a realizar inspecciones continuas de código de sus proyectos. La herramienta analiza 30+ lenguajes de programación diferentes y se integra en su canalización de CI y plataforma DevOps para garantizar que su código cumpla con los estándares de alta calidad[13].

Figura 6.14: Ciclo que propone SonarQube



Fuente: Página oficial de SonarQube <https://docs.sonarqube.org/latest/>

Con esta herramienta se pretende llegar a tener la menor cantidad de Code Smells¹ posibles, así como que la deuda técnica² sea la menor posible. También evitar la aparición de bugs, o vulnerabilidades en el código.

6.8. Control de versiones

6.9. Git

La información de este apartado se ha obtenido desde la página de Atlassian [<https://www.atlassian.com/git>]

Git es un sistema de control de versiones gratuito y de código abierto, creado originalmente por Linus Torvalds en 2005. Con Git, cada desarrollador tiene el historial completo de su repositorio de código localmente. Esto hace que la clonación inicial del repositorio sea más lenta, pero las operaciones posteriores, como confirmar, diferenciar, fusionar y registrar, son mucho más rápidas.

Figura 6.15: Logo de Figma



Fuente: Iconduck [<https://iconduck.com/icons/27401/git>]

Git también tiene un excelente soporte para bifurcar, fusionar y reescribir el historial del repositorio, lo que ha dado lugar a muchos flujos de trabajo y herramientas innovadoras y potentes. Las solicitudes de incorporación de cambios son una de esas herramientas populares que permiten a los equipos colaborar en ramas de Git y revisar de manera eficiente el código de los demás. Git es el sistema de control de versiones más utilizado en el mundo actual y se considera el estándar moderno para el desarrollo de software.

Las órdenes básicas de Git son:

- `git init`: crea un nuevo repositorio de Git vacío en el directorio actual.
- `git clone`: clona un repositorio existente en un nuevo directorio.
- `git add`: agrega cambios al área de preparación (staging area) para que estén listos para ser confirmados.

¹El **code smell** es una indicación de la mala calidad del código. Si hay code smells, quien lea el código tendrá la sensación de que algo está mal. Este se soluciona refactorizando código.[14]

²La **deuda técnica** es el costo del retrabajo adicional causado por la elección de la solución más rápida en lugar de la más efectiva.[15]

- `git commit`: crea un nuevo commit (instantánea) con los cambios agregados al área de preparación y agrega un mensaje de confirmación que describe los cambios.
- `git status`: muestra el estado actual del repositorio, incluyendo los cambios sin confirmar y los archivos sin seguimiento.
- `git log`: muestra una lista de todos los commits en orden cronológico inverso.
- `git pull`: actualiza el repositorio local con los cambios más recientes del repositorio remoto.
- `git push`: envía los cambios locales al repositorio remoto.
- `git branch`: muestra una lista de todas las ramas en el repositorio.
- `git checkout`: cambia a otra rama o commit.

6.10. GitKraken

GitKraken es una herramienta de gestión de versiones de código que ofrece una interfaz visual y fácil de usar para trabajar con Git. Es una aplicación de escritorio que permite a los desarrolladores y equipos de desarrollo colaborar en proyectos de software de manera eficiente.

Con GitKraken, se pueden realizar operaciones comunes de Git, como hacer commits, fusionar ramas, crear y clonar repositorios, y gestionar conflictos de fusión.

Además, cuenta con características adicionales, como la posibilidad de integrarse con plataformas de gestión de proyectos como Trello, la capacidad de ver el historial de cambios en el código, y la opción de visualizar y comparar ramas de Git.

Grandes empresas hacen uso de esta herramienta, empresas tales como Netflix, Philips, Amazon, Unity y Disney entre otras.

Figura 6.16: Logo de Figma



Fuente: Iconduck [<https://iconduck.com/icons/27407/gitkraken>]

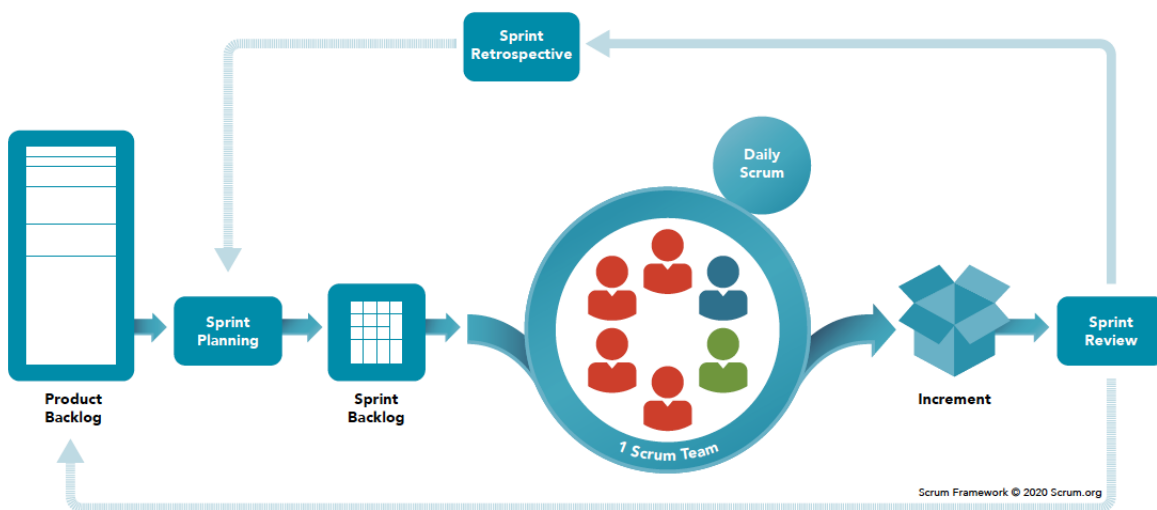
Capítulo 7

Metodología

7.1. Marco de Trabajo SCRUM

Para la creación de la aplicación Web3 se ha seguido un marco de trabajo inspirado en **SCRUM** para un desarrollo ágil. Este marco de trabajo está fundamentado por la existencia de etapas de tiempo previamente establecidas donde se enfoca la carga de trabajo, consiguiendo un producto mínimamente viable en cada una de ellas. Estas etapas de tiempo se conocen por el nombre de Sprints, y cada uno de ellos se divide en diferentes fases: planificación del Sprint, realización del Sprint, con reuniones diarias, y retrospectiva del Sprint. En la Figura 7.1 se muestra el flujo de trabajo que se suele seguir en la metodología SCRUM.

Figura 7.1: Metodología SCRUM



Fuente: [www.scrum.org \[https://www.scrum.org/learning-series/what-is-scrum\]](https://www.scrum.org/learning-series/what-is-scrum)

Planificación del Sprint: En esta fase, se definen los objetivos del sprint, identifican las tareas necesarias para cumplir esos objetivos y determinan la cantidad de trabajo que se

puede realizar en el sprint.

Sprint: Esta es la fase en la que se realiza el trabajo. Durante el sprint, se trabaja en las tareas definidas en la planificación del sprint para lograr los objetivos establecidos.

Reunión diaria: En esta fase, el equipo Scrum se reúne diariamente para revisar el progreso del trabajo, identificar cualquier problema y hacer ajustes necesarios para alcanzar los objetivos del sprint.

Revisión del Sprint: Al final del sprint, el equipo Scrum se reúne para revisar el trabajo completado y demostrar los resultados a los interesados. Esto ayuda a identificar los logros y también a las áreas donde se pueden mejorar.

Retrospectiva del Sprint: En esta fase, el equipo Scrum reflexiona sobre el sprint anterior y discute qué se hizo bien y qué se puede mejorar en el próximo sprint.

Planificación de lanzamiento: Si se han completado múltiples sprints, el equipo Scrum se reúne para planificar la entrega del producto final al cliente.

A la hora de hacerlo, al no tener un "equipo", sino ser únicamente una persona, la **planificación** del sprint se realizaba mediante el uso de Trello[6.5], añadiendo tareas, y colocando sus respectivos "tests", que en su mayoría son validaciones por parte de personas externas.

La **realización del sprint** se realizaba en diferentes apartados, los cuales son:

La parte de **diseño** realizado en Figma[Sección 6.6]

- Diseño de componentes
- Diseño de páginas con los componentes diseñados

La parte de **desarrollo** realizado en Figma[Sección 6.4] haciendo uso de las distintas herramientas antes mencionadas[Capítulo 6].

- Desarrollo de la página para poder colocar los componentes
- Desarrollo de los componentes e insertarlos de forma adecuada en la página

La parte de **validación** realizado con Lighthouse[Sección 6.7.1] y SonarQube[Sección 6.7.2]

- Siguiendo el consejo antes mencionado de SonarQube, se pasaba el test antes de hacer un *push*, comprobando la posible existencia de Code Smells, bugs, vulnerabilidades, etc.
- A la hora de pasar la validación de Lighthouse se hizo cada vez que se hacía un *merge* a la rama principal, se pasaba este test para ver las posibles fallas de accesibilidad que tenía la página en ese momento y poder solucionarlas.

Capítulo 8

Análisis y Diseño

Capítulo 9

Desarrollo

Capítulo 10

Evaluación y resultados

Capítulo 11

Conclusiones y trabajo futuro

CAPÍTULO OBLIGATORIO

Resultados, grado de consecución de los objetivos, posibles extensiones

Bibliografía

- [1] IBM. ¿tecnología blockchain? <https://www.ibm.com/es-es/topics/blockchain>. [Online; accedido a 30/April/2023].
- [2] BBC News Mundo. Qué son los nft y por qué están valorados en millones de dólares. <https://www.bbc.com/mundo/noticias-56502251>, March 2021. [Online; accedido a 01/May/2023].
- [3] Moralis. What is a web3 wallet? – web3 wallets explained. https://moralis.io/what-is-a-web3-wallet-web3-wallets-explained/?ref=morioh.com&utm_source=morioh.com, January 2022. [Online; accedido a 01/May/2023].
- [4] Open Bootcamp. ¿qué es un framework y qué tipos hay? <https://open-bootcamp.com/aprender-programar/que-es-un-framework>. [Online; accedido a 01/May/2023].
- [5] CERN. A short history of the web. <https://home.cern/science/computing/birth-web/short-history-web>. [Online; accedido a 02/May/2023].
- [6] Heather Hall. Web 2.0 explained: Everything you need to know. <https://history-computer.com/web-2-0/>, November 2022. [Online; accedido a 02/May/2023].
- [7] Sean Yang and Max Li. Web3.0 data infrastructure: Challenges and opportunities. <https://ieeexplore.ieee.org/document/10110018>, April 2023. [Online; accedido a 02/May/2023].
- [8] Yúbal Fernández. Que son los nft y cómo funcionan. <https://www.xataka.com/basics/que-nft-como-funcionan>, July 2022. [Online; accedido a 02/May/2023].
- [9] Nelson Rodriguez. 35+ ejemplos de web 3.0 y como blockchain está cambiando la web. <https://101blockchains.com/es/web-3-0/>, 12 2018. [Online; accedido a 12/05/2023].
- [10] Truust. Peer to peer (p2p): Qué es y qué ventajas tiene. <https://truust.io/es/blog/peer-peer-p2p-que-es-y-que-ventajas-tiene/>. [Online; accedido a 12/05/2023].
- [11] Blogdecentralab. Qué es alchemy. <https://decentralab.tech/blog/que-es-alchemy/>, 03 2023. [Online; accedido a 13/05/2023].
- [12] developers.google.com/web. Lighthouse. <https://chrome.google.com/webstore/detail/lighthouse/blipmdconlkpinefehnmjammfjpmpbjk?hl=es>. [Online; accedido a 13/05/2023].

- [13] SonarQube. Sonarqube documentation. <https://docs.sonarqube.org/latest/>. [Online; accedido a 13/05/2023].
- [14] Desarrollo web. Code smell. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/code-smell/>, 11 2022. [Online; accedido a 13/05/2023].
- [15] Team Asana. Qué es la deuda técnica y cómo saldarla (con ejemplos). <https://asana.com/es/resources/technical-debt>, 07 2022. [Online; accedido a 13/05/2023].