

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN
TECNOLOGÍAS DE LA INFORMACIÓN



"ANÁLISIS Y DESARROLLO DE UNA
APLICACIÓN PARA EL MERCADO DE
NFTS EN DISPOSITIVOS MÓVILES"

TRABAJO FIN DE GRADO

Diciembre –
2023

AUTOR: Samuel Candela Peral

DIRECTOR/ES: Antonio Peñalver Benavent



AGRADECIMIENTOS Y DEDICATORIAS

Este trabajo final de grado es un reflejo de cómo puedes encontrar una pasión dentro de otra pasión al finalizar la etapa más dura de mi vida, poniendo punto final a esta aventura.

La elaboración de este trabajo ha sido el empujón que necesitaba para descubrir y adentrarme en el mundo del desarrollo de aplicaciones móviles, descubriendo así un nuevo lenguaje y una nueva metodología de programación.

Gracias, en primer lugar, a mis padres por nunca haberse rendido conmigo y haber hecho hincapié en que lo mejor para mi futuro era estudiar, por hacer que nunca me haya faltado nada, y lo más importante disciplina y trabajo duro.

Gracias a mi tutor Antonio Peñalver por las ayudas proporcionadas en cuanto a la elección del tema, consejos y facilidades.

Gracias también, a aquellos familiares que no han podido ver hasta donde he conseguido llegar, pero gracias a su recuerdo, siempre parto con un pequeño empujón extra.

Y finalmente, sobre todo, gracias a mí mismo por nunca haberse rendido a pesar de todas las dificultades añadidas que ha habido por este largo camino, haciéndole ver a aquel adolescente de 16 años que no quería seguir estudiando depresivo por el fútbol, que, hoy, hemos conseguido un mejor y prometedor futuro.



RESUMEN

En estos últimos años las criptomonedas y los NFTs han alcanzado una importancia significativa en un mundo cada vez más digitalizado en busca de innovaciones y progresos. Hoy en día, estos conceptos están muy presentes en las nuevas generaciones y representan un futuro inminente.

Los NFTs, basados en la tecnología de Blockchain, han surgido como activos digitales únicos y auténticos, extendiendo su influencia a una multitud de campos, desde el arte digital hasta bienes raíces y videojuegos.

Más allá de las inversiones, están reescribiendo la cultura digital, permitiendo a creadores y coleccionistas explorar nuevas formas de expresión y propiedad. En este contexto, los NFTs representan un cambio fundamental en la comprensión de la propiedad digital y continuarán influyendo en el mundo digital en el futuro. Se trata de elementos diferenciales que puede poseer cada una de las personas, convirtiéndose estas en únicas.

Debido a la inmediatez de este cambio, he decidido enfocar mi trabajo de fin de grado en el desarrollo de un mercado de NFTs el cual sea accesible para todo el mundo, donde cualquiera pueda comprar arte digital con la mayor facilidad posible, profesionalizando mi gran interés por el desarrollo de aplicaciones móviles.



ÍNDICE

INTRODUCCIÓN	14
1.1 ¿QUE SON LOS NFTS Y PARA QUÉ SIRVEN?	14
1.2 ¿CÓMO FUNCIONAN LOS NFTS?	15
1.3 ¿QUÉ ES UN MERCADO DE NFTS?	16
1.4 ¿QUÉ SON LAS COLECCIONES DE NFT?	17
1.5 ¿CUÁL ES EL PRINCIPAL PROBLEMA?	19
1.6 MOTIVACIONES	21
1.7 ¿QUÉ SE PRETENDE CONSEGUIR?	22
ESTADO DE LA CUESTIÓN	24
2.1 POPULARIZACIÓN DE LOS NFTS	24
2.2 MERCADOS DE NFTS	25
2.3 BLOCKCHAINS Y COMISIONES	28
OBJETIVOS DEL PROYECTO	31
3.1 OBJETIVOS GENERALES	31
3.2 OBJETIVOS TÉCNICOS	32
3.3 OBJETIVOS DIDÁCTICOS	32
3.4 OBJETIVOS PERSONALES	33
3.5 NO OBEJTIVOS	33
HIPÓTESIS DE TRABAJO	35
4.1 ANDROID STUDIO	36
4.2 KOTLIN	37
4.3 METAMASK	38

4.4	FIGMA	39
4.5	GIT Y GITHUB - SOURCETREE	40
4.6	CLEAN ARCHITECTURE	41
4.6.1	DEFINICIÓN	41
4.6.2	ESTRUCTURA	43
4.6.3	VENTAJAS	46
4.6.4	DESVENTAJAS	47
4.7	MVVM	48
4.7.1	DEFINICIÓN	48
4.7.2	IMPLEMENTACIÓN	50
4.8	DEPENDENCIAS	53
4.8.1	CLOUDINARY	55
4.8.2	ANDROID X	55
4.8.3	AWS AMPLIFY ANDROID	56
4.8.4	GSON	57
4.8.5	CONCLUSIÓN	57
METODOLOGÍA Y RESULTADOS		59
5.1	CICLO DE VIDA DEL PROYECTO	59
5.1.1	REQUERIMIENTOS	61
5.1.2	ANÁLISIS Y DISEÑO	62
5.1.2.1	DIAGRAMA DE GANTT	62
5.1.2.2	ACTORES Y CASOS DE USO	67
5.1.2.2.1	ACTORES	69
5.1.2.2.2	CASOS DE USO	70
5.1.2.3	DIAGRAMAS DE ACTIVIDAD	77
5.1.2.4	DISEÑO INTERFAZ	81
5.1.2.4.1	HOME	83
5.1.2.4.2	EXPLORE	84
5.1.2.4.3	PROFILE	86
5.1.2.4.4	BOTON +	87
5.1.2.4.5	FEED	87
5.1.3	IMPLEMENTACIÓN Y PRUEBAS UNITARIAS	88
5.1.3.1	PRUEBAS Y ERRORES	88

5.1.4	VALIDACIÓN Y PRUEBAS	91
5.1.5	IMPLANTACIÓN Y MATENIMIENTO	92
CONCLUSIONES Y PROPUESTAS		94
6.1	CONCLUSIONES	94
6.2	TRABAJOS FUTUROS	95
BIBLIOGRAFÍA		97
7.1	FUENTES BIBLIOGRÁFICAS	97

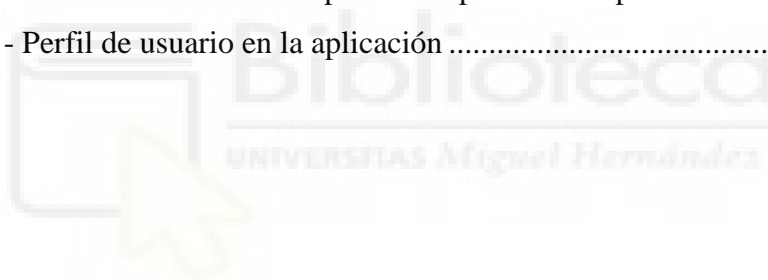




ÍNDICE DE FIGURAS

Ilustración 1- Ilustración de token no fungible - NFT.....	14
Ilustración 2 - Que incluye la Blockchain y su conectividad	15
Ilustración 3 - Muestra de NFTs de una misma colección.	17
Ilustración 4 - Recreación de una galería de NFTs en un museo.	18
Ilustración 5 - Cibercriminales y los peligros de los NFTs.....	20
Ilustración 6 - Grafica del crecimiento del número de billeteras activas en el mercado de NFTs	24
Ilustración 7 - Grafico de ventas semanales de los principales mercados de NFTs.....	26
Ilustración 8 - Logotipo de OpenSea.....	26
Ilustración 9 - Logotipo de Atomic Market.....	27
Ilustración 10 - Logotipo de Rarible	27
Ilustración 11 - Logotipo de SuperRare	28
Ilustración 12 - Logotipo de Kotlin.....	37
Ilustración 13 - Ilustración del depósito de criptomonedas en la billetera Metamask. ..	38
Ilustración 14 - Demostración del flujo de pantallas y su diseño en Figma.....	39
Ilustración 15 - Representación del uso de GitHub y Sourcetree combinados	40
Ilustración 16 - Esqueleto y flujo de una arquitectura limpia.....	42
Ilustración 17 - Representación de las capas que componen Clean Architecture con sus componentes y relaciones.....	44
Ilustración 18 - Ilustración que representa la arquitectura simplificada de Clean Architecture.....	45
Ilustración 19 - Esqueleto y flujo de MVVM. Direcciones de Data y Data Request.....	49
Ilustración 20 - Organización paquetes del proyecto.	50
Ilustración 21 - Representación división de paquetes internos del proyecto (Core, Features, Services, Utils).....	51
Ilustración 22 - División de los paquetes por cada pantalla y sus funciones.....	51
Ilustración 23 - Representación de división por capas de MVVM.....	52
Ilustración 24 - Contenido de los features y las clases para llamadas al servidor.....	52
Ilustración 25 - Archivo Gradle del proyecto.....	54

Ilustración 26 - Logotipo de Cloudinary.	55
Ilustración 27 Esqueleto y funcionamiento de la API REST.	56
Ilustración 28 - Ciclo de vida en cascada seguido.....	60
Ilustración 29 - Tareas diagrama de Gantt.....	63
Ilustración 30 - Vista en semanas del diagrama de Grant.	65
Ilustración 31 - Ilustración en cascada del Diagrama de Grant.....	66
Ilustración 32 - Casos de uso del usuario	69
Ilustración 33 - Ilustración Diagrama de actividad inicio de aplicación	78
Ilustración 34 - Ilustración Diagrama de actividad consulta información perfil.....	79
Ilustración 35 - Ilustración Diagrama de actividad consulta del Home	80
Ilustración 36 - Inicio sesion en la aplicación	81
Ilustración 37 - Navegación de acciones internas de la aplicación.	82
Ilustración 38 -Home de la aplicación.....	83
Ilustración 39 - Sección Explore de la aplicación.	84
Ilustración 40 - Listado usuarios en el apartado explore de la aplicación.....	85
Ilustración 41 - Perfil de usuario en la aplicación	86





CAPÍTULO 1

INTRODUCCIÓN

1.1 ¿QUE SON LOS NFTS Y PARA QUÉ SIRVEN?

Los NFTs [1], abreviatura de "Token No Fungible", representan una innovación en el mundo digital. Se trata de activos digitales únicos y certificados que utilizan la tecnología Blockchain [2] para garantizar su autenticidad y singularidad. La Blockchain es una red descentralizada que registra y verifica transacciones de forma segura, sin necesidad de intermediarios. Esto significa que los usuarios tienen un control total sobre las transacciones realizadas [3].

Con Blockchain, se elimina completamente al intermediario de la ecuación y a su vez encripta la información manteniendo así la privacidad y seguridad.



Ilustración 1- Ilustración de token no fungible - NFT

A diferencia de las criptomonedas, como Bitcoin [4] o Ethereum [5], que son fungibles y pueden ser intercambiadas entre sí, los NFTs son no fungibles, lo que significa que cada uno de ellos es único e irrepetible. No se pueden duplicar, lo que les proporciona su valor como objetos digitales únicos e infalsificables.

La característica distintiva de estos activos radica en la capacidad de certificar la propiedad y autenticidad de activos digitales, como obras de arte, música, videojuegos y otros contenidos digitales. Estos tokens se utilizan para representar la propiedad digital, permitiendo a los usuarios comprar, vender y poseer activos digitales únicos en el mundo virtual [6].

1.2 ¿CÓMO FUNCIONAN LOS NFTS?

Los NFTs funcionan mediante tecnología Blockchain o de cadena de bloques, usando la misma tecnología de las criptomonedas, que funcionan mediante una red de ordenadores descentralizada, con bloques o nodos enlazados y asegurados usando criptografía. Cada uno de estos bloques enlaza a un bloque previo, con una fecha y datos de transacciones, y por diseño son resistentes a la modificación de datos [6].

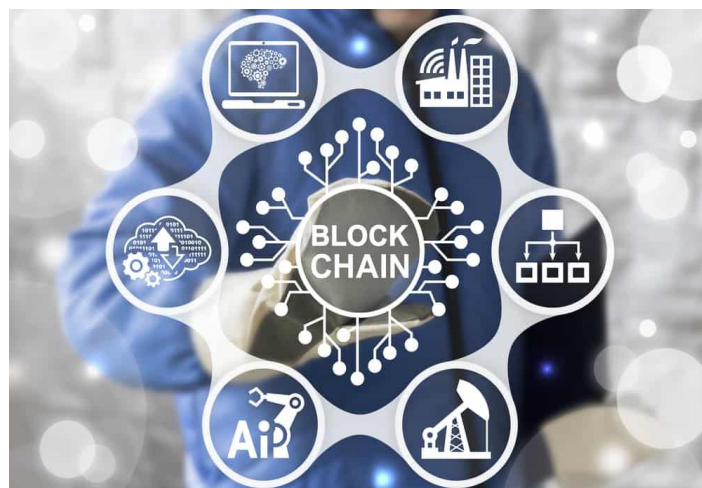


Ilustración 2 - Que incluye la Blockchain y su conectividad

A los NFTs, se les asigna una especie de certificado digital de autenticidad, una serie de Metadatos [7] que no se van a poder modificar, garantizando con estos metadatos su autenticidad, el valor de partida y todas las adquisiciones o transacciones que se hayan hecho, además de su autor.

Al utilizar la tecnología Blockchain y estos certificados digitales, los NFTs permiten a los creadores y coleccionistas rastrear y verificar la autenticidad y propiedad de activos digitales en un mundo cada vez más digitalizado.

Esto quiere decir que, si se compra un contenido digital Tokenizado [8] como NFT, en todo momento siempre habrá constancia del primer valor que tenía, y de por cuánto se ha comprado. Es como cuando se compra un cuadro y se lleva un registro de por dónde se mueve [9].

1.3 ¿QUÉ ES UN MERCADO DE NFTS?

Un mercado de NFTs es una plataforma en línea que permite a los usuarios emitir, comprar y vender activos digitales únicos. Para utilizar estos mercados, se necesita disponer previamente de un Cripto-Monedero [10] compatible con el Blockchain que emplee dicho mercado. Además, también se necesita disponer de criptomonedas en tu monedero para poder acuñar, comprar o vender.

Cuando se visita un mercado, se puede ver qué NFTs están a la venta en ese momento, además de otros detalles como cuándo fueron emitidos o qué ofertas han hecho otros usuarios sobre ellos, por ejemplo.

Estos mercados funcionan como plataformas en las que los creadores pueden subir sus NFTs para su venta, establecer precios y condiciones, y los compradores interesados pueden explorar y buscar activos digitales que les interesen. Cuando

un comprador adquiere uno en un mercado, se registra la transacción en la Blockchain, lo que garantiza la autenticidad y la propiedad del activo digital [11].

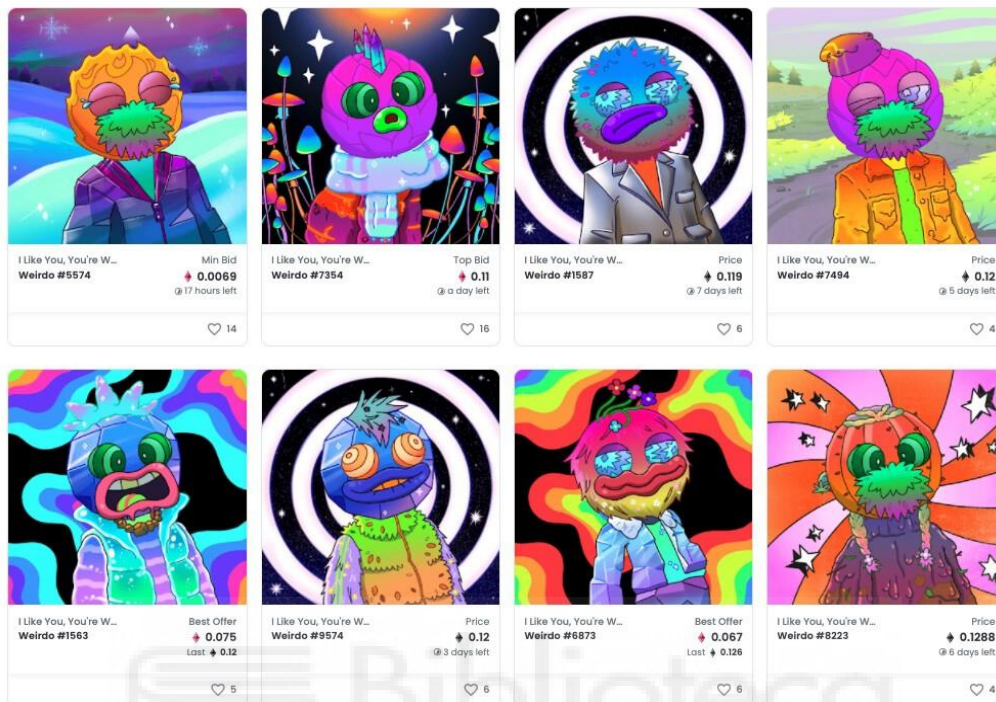


Ilustración 3 - Muestra de NFTs de una misma colección.

1.4 ¿QUÉ SON LAS COLECCIONES DE NFT?

Una colección de NFTs se compone de varios NFTs que tienen una estrecha relación entre ellos y están desarrollados por el mismo artista o grupo de artistas, compartiendo ciertas características o propiedades comunes. Estas colecciones pueden estar ya completas al ser puestas en circulación, o sus creadores pueden añadir nuevos ítems a la colección en una fecha posterior [11].

Cada NFT dentro de una colección sigue siendo único e irrepetible, pero todos los de una misma colección suelen estar relacionados de alguna manera.

Aquí hay algunos ejemplos de lo que podrían ser las colecciones de NFTs:

Colecciones de arte: Un artista digital puede crear una serie de NFTs que representen diferentes obras de arte digitales. Cada uno de estos podría ser una pieza de arte única, pero todos estarían dentro de la misma colección del artista.

Colecciones de juegos: En el mundo de los videojuegos, se utilizan para representar elementos de juego únicos, como personajes, objetos o Skins [12]. Estos pueden agruparse en colecciones relacionadas con un juego en particular.

Colecciones temáticas: Algunas colecciones están relacionadas con un tema o concepto específico. Por ejemplo, podría existir una colección de gatos digitales o una colección que representen momentos históricos.

Colecciones de coleccionistas: Los coleccionistas de NFTs pueden agrupar sus adquisiciones en colecciones personales basadas en sus intereses y preferencias. Estas colecciones pueden ser públicas o privadas.

Colecciones de edición limitada: Los NFTs a menudo se lanzan en ediciones limitadas, y todas las copias de una misma edición se consideran parte de una colección específica.



Ilustración 4 - Recreación de una galería de NFTs en un museo.

Las colecciones de NFTs pueden tener valor tanto para los creadores como para los coleccionistas. Los creadores pueden establecer un tema o una narrativa en torno a su colección, lo que puede aumentar su atractivo y valor en el mercado. Los coleccionistas, por otro lado, pueden buscar y comprar los activos digitales que se ajusten a sus intereses personales o estrategias de inversión. En estos mercados, a menudo puedes explorar y buscar NFTs por colección para encontrar activos digitales que te interesen [13].

1.5 ¿CUÁL ES EL PRINCIPAL PROBLEMA?

El mercado de los NFTs y los NFTs en sí mismos han experimentado un rápido crecimiento y popularidad en los últimos años, pero también han enfrentado una serie de desafíos y problemas. Algunos de los problemas actuales asociados con los NFTs y su mercado incluyen:

Escepticismo y burbuja especulativa: Algunos críticos consideran que el mercado de los NFTs está inflado y que existe una burbuja especulativa en torno a estos activos.

Dificultad para evaluar el valor: Determinar el valor real de un NFT puede ser complicado. A menudo, este valor se basa en la demanda del mercado y en factores subjetivos como la reputación del creador, lo que puede hacer que los inversionistas enfrenten riesgos significativos.

Derechos de autor y plagio: La naturaleza descentralizada y anónima de los NFTs ha llevado a problemas de derechos de autor y plagio. Se han vendido NFTs que infringen derechos de propiedad intelectual sin el consentimiento del creador original.

Estafas y fraudes: Como en cualquier mercado en línea, existen estafadores que intentan engañar a los compradores y vendedores. Los compradores pueden encontrarse con NFTs falsos o no auténticos, y los vendedores pueden ser víctimas de estafas de pago.

Problemas de escalabilidad: Las redes Blockchain actuales, como Ethereum, han experimentado problemas de congestión y altas tarifas de transacción debido a la creciente demanda. Esto ha generado preocupaciones sobre la escalabilidad de la tecnología.

Falta de regulación: La falta de regulación en el espacio de los NFTs ha llevado a un entorno poco regulado y propenso a abusos. Esto ha generado preocupaciones sobre la protección del consumidor y la seguridad financiera.

Dificultad de compra: Es cierto que comprar un NFT en un mercado no es una tarea difícil, pero requiere de un cierto conocimiento previo algo complejo y poco intuitivo que dificulta que estos sean accesibles para todo el mundo.

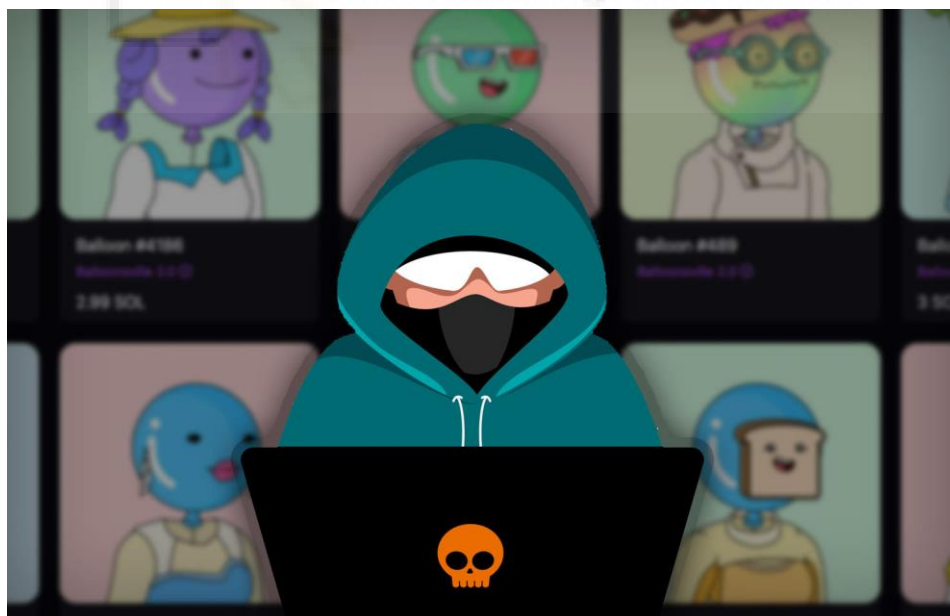


Ilustración 5 - Ciberdelincuentes y los peligros de los NFTs

Es importante tener en cuenta que este mercado es relativamente nuevo y sigue evolucionando. A medida que estos problemas se vuelven más evidentes, es probable que se desarrollen soluciones y regulaciones para abordarlos. Los inversores y participantes en dicho mercado deben ser conscientes de estos desafíos y realizar una investigación exhaustiva antes de participar en transacciones [13].

1.6 MOTIVACIONES

La principal motivación para el desarrollo de este proyecto surge de la fusión de mi interés en la creación de aplicaciones móviles en el ámbito profesional y mi atractivo por las inversiones y la tecnología Blockchain y NFTs.

Este proyecto no es más que la materialización de mi deseo por explorar y comprender la complejidad de la creación de una aplicación desde cero, integrando elementos y conceptos como Blockchain y APIs externas.

El principal objetivo es realizar una investigación y desarrollo de un proyecto que incluya complejidades técnicas y conceptuales asociadas a la tecnología Blockchain, así como la integración de NFTs, Metamask, y AWS, las cuales se emplean en el ámbito profesional avanzado, constituyendo así un desafío enriquecedor que me permita explorar y superar obstáculos consolidando mis habilidades y conocimiento en el desarrollo de software.

A su vez quiero buscar una mejora en la sencillez de la interfaz de otras aplicaciones de mercados de NFTs priorizando la simplificación y optimización de dicha interfaz mejorando a su vez la accesibilidad y usabilidad del usuario, donde se proporciona una experiencia fluida e intuitiva.

1.7 ¿QUÉ SE PRETENDE CONSEGUIR?

El principal objetivo de este proyecto es desarrollar un mercado de NFTs que sea accesible para todos los usuarios a través de una aplicación móvil. Se pretende garantizar que este mercado sea accesible a un público amplio y diverso.

Se trata de resolver el problema de accesibilidad para todos los públicos, implementando una navegación y una simplificación únicas. La interfaz de la aplicación se dividirá en sectores aclarativos para que el usuario pueda navegar y consultar de manera simple cualquier operación relacionada con el mercado de NFTS.

Esta estructuración facilitará la comprensión y utilización de la aplicación, asegurando que incluso aquellos menos familiarizados con el entorno de NFTs puedan participar de manera efectiva

Se llevará a cabo la creación de una primera versión y diseño de aplicación para conocer a fondo como es el desarrollo de una aplicación completa. Esta fase inicial del desarrollo tiene como objetivo proporcionar una comprensión profunda de los procesos y desafíos asociados con la creación de una aplicación completa.



CAPÍTULO 2

ESTADO DE LA CUESTIÓN

2.1 POPULARIZACIÓN DE LOS NFTS

Desde 2018 la capitalización de mercado de los NFTS ha sufrido grandes aumentos y es en el último cuarto de 2021 donde su popularización y capital se dispara en más de un 3000 % llegando a máximos históricos. Es a partir de ese momento donde todo el mercado se populariza y se comienzan a realizar miles de compraventas al día y la gente comienza a hablar de estas inversiones [14].

Con este auge comienza la necesidad de continuar expandiendo esta tendencia con aplicaciones web y móviles para la comercialización de las nuevas colecciones.

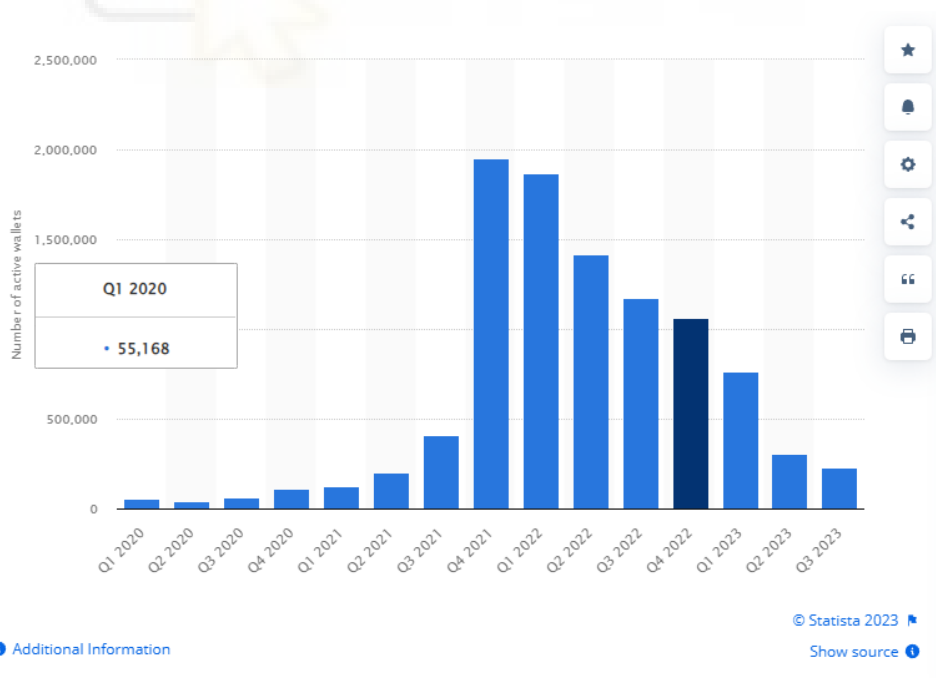


Ilustración 6 - Grafica del crecimiento del número de billeteras activas en el mercado de NFTs

A su vez también podemos observar una gran caída, no tan agresiva como la subida que tuvo esta, pero si con mucho ritmo. Esto se debe a varios factores que han afectado la relación usuario-mercado NFTs, como han sido la gran bajada de las criptomonedas como fue Bitcoin arrastrando a todas las demás, el cual alcanzo un pico de 66.000 dólares y bajo en más de un 45 % en un mes, llegando a descender a 16.000 dólares finalmente como pico más bajo. Esto provoco que toda la gente con un conocimiento medio en este sector perdiera toda su confianza debido a la gran volatilidad de este mercado, perdiendo así muchos inversores [14].

Además, una de las grandes razones de esta caída fue que el gran aumento de este mercado impulsara la creación de nuevas estafas, donde se creaban colecciones de NFTs con grandes promesas y volumen de mercado y acaban siendo estafas donde los creadores originales de la colección robaban dinero a los usuarios.

Un gran número de personas acabaron afirmando que les parecía que los NFTs resultaban una estafa y estaban descontentos y no llegaban a entender su valor monetario.

2.2 MERCADOS DE NFTS

A raíz de este crecimiento masivo, se han desarrollado varios portales web que permiten la compraventa de estos activos, actualizándose a la actual Web 3.0 [11], como pueden ser OpenSea [15] el cual lidera el sector, AtomicMarket [16], Rarible [17], SuperRare [18] y varios más [19].

Marketplace: Count of Sales - 7 Days Trailing Average, 2021

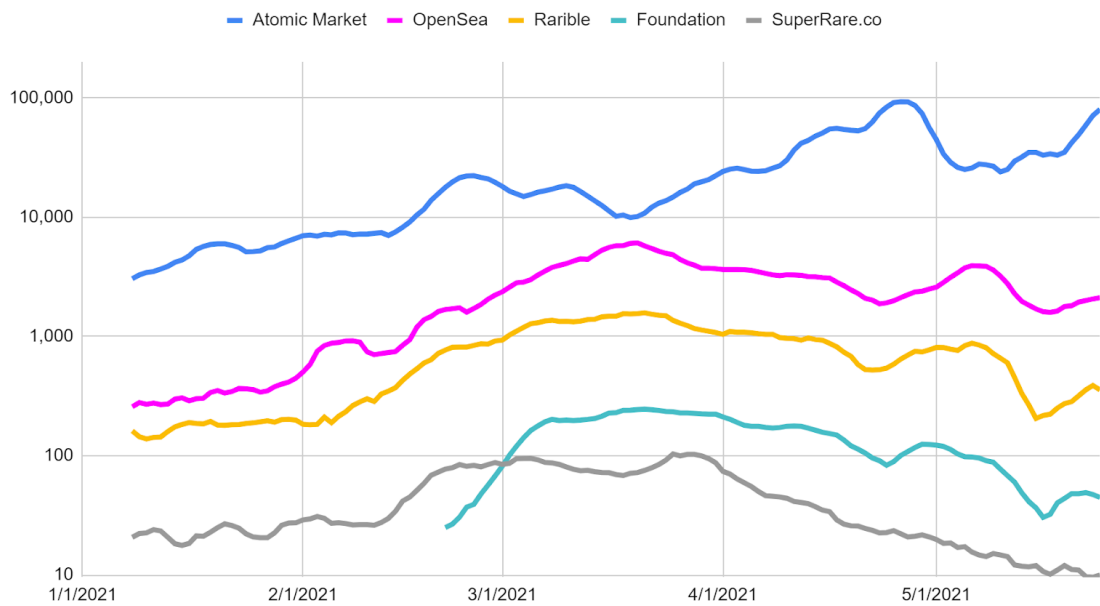


Ilustración 7 - Grafico de ventas semanales de los principales mercados de NFTs

A continuación, vamos a ver una breve descripción de estos diferentes mercados de NFTs para ver sus características y en que son diferenciales:

OpenSea: Este portal es el padre de los demás, siendo este el más antiguo y establecido del mundo. Tiene la mayor selección de NFTs en el mercado y posee comisiones bajas y honorarios para los creadores.



Ilustración 8 - Logotipo de OpenSea

Atomic Market: Posee una interfaz fácil de usar y permite a los usuarios crear y vender sus propios NFTs. Este mercado destaca por su gran seguridad empleando la cadena de bloques WAX, proponiendo así su propio token.



Ilustración 9 - Logotipo de Atomic Market

Rarible: Es una plataforma más enfocada a un público nuevo en la creación de NFTs, ya que posee también su propio token en el que está basado el portal. Aborda limitaciones del mercado de la propiedad intelectual, como el papeleo y licencias, centrándose en un uso más autónomo



Ilustración 10 - Logotipo de Rarible

SuperRare: Posee un mercado de NFTs centrado solamente en la Blockchain de Ethereum centrándose más en el arte digital único y de alta calidad, teniendo un sistema de subastas que permite a los compradores pujar por NFTs muy exclusivos de dicho portal.



Ilustración 11 - Logotipo de SuperRare

El principal problema común que considero que tienen estos mercados es la carencia de intuitiva navegación en sus portales web y aplicaciones móvil. Se requiere algo de conocimiento previo y no es tan fácil descubrir colecciones.

Además, las aplicaciones móviles tienen un funcionamiento distinto a sus portales web, llegando a confundir al usuario promedio.

Cabe destacar, además, que solo un par ofrecen facilidades al creador de arte digital.

2.3 BLOCKCHAINS Y COMISIONES

Los NFTS, en su gran mayoría, están desarrollados en las Blockchains principales como Ethereum (líder del sector), Binance Smart Chain [20], Polygon [21], Solana [22], entre otras.

Sin embargo, aunque principalmente las compras de estos se realizan en Ethers, también hay NFTs que han de ser comprados por la criptomoneda de las Blockchains comentadas anteriormente. Anteriormente habíamos visto también como hay ciertos portales web de compraventa que han apostado por la unificación de todos los NFTs en la misma tecnología Blockchain.

Esta diversificación conlleva a una falta de centralización total de todo el mercado de NFTs y obliga al usuario a poseer diferentes criptomonedas o realizar compras previas, lo que provoca pérdidas en comisiones.

Es crucial señalar que estos mercados de NFTs aplica un alto porcentaje de retención sobre cada compra para los artistas llegando hasta un 10 % de cada compra que se realiza con cualquier NFT de la colección. En otras palabras, el artista que realiza la subida de la colección por cada transacción de un NFT de su colección se lleva un 10 % del total, a esto también se le añade cada porcentaje de criptomoneda que se lleva el portal web/aplicación.

Este sistema de comisiones representa un costo significativo para el comprador ya que cada transacción de compraventa de un NFT que posee implica una pérdida de un porcentaje del total del valor de este y resulta una forma de lucrarse para el creador de dicha colección.

Este es un aspecto financiero que es esencial tenerlo en cuenta para la evaluación de la viabilidad y sostenibilidad económica del mercado de NFTs.



CAPÍTULO 3

OBJETIVOS DEL PROYECTO

Se ha realizado una clasificación de los objetivos del proyecto para tener una estructura más clara de dichos objetivos en cada ámbito.

La clasificación que se ha realizado ha sido de:

- **Objetivos generales.**
- **Objetivos técnicos.**
- **Objetivos didácticos**
- **Objetivos personales**
- **No objetivos.**



3.1 OBJETIVOS GENERALES

A continuación, se van a establecer los objetivos generales de este proyecto, los cuales serían los siguientes:

- Desarrollar el prototipo y diseño de una aplicación móvil funcional en la cual se pueden consultar como esta estará distribuida y acceder a la vista de NFTs y colecciones de estos, así como sus precios y descripciones.
- Diseñar una interfaz limpia e intuitiva que permita ser utilizada durante el uso de la aplicación para lograr objetivos propuestos al entrar en la aplicación priorizando la sencillez para el uso de cualquier público.

3.2 OBJETIVOS TÉCNICOS

En cuanto a objetivos técnicos se han determinado los siguientes puntos:

- Realizar una conexión directa con una cartera existente de Metamask para realizar un inicio de sesión donde queda registrada la dirección de billetera con la que se realizara compraventas de NFT.
- Desarrollo de la aplicación para una primera versión en Android, incluyendo en esta la interfaz final y la funcionabilidad de la aplicación.
- Desarrollar una interfaz fácil e intuitiva que permita al usuario sin experiencia poder usar perfectamente la aplicación sin necesidad de consultas previas.

3.3 OBJETIVOS DIDÁCTICOS

Los objetivos didácticos han sido varios durante la creación del proyecto, pero se ha realizado una numeración de cuáles han sido los principales:

- Realizar un control de versiones, empleando Git y GitHub.
- Uso de APIs externas, como AWS.
- Uso de peticiones al servidor que devuelve datos en forma de JSON [23].
- Transformación de datos JSON en Objetos.
- Explicar los problemas encontrados durante el desarrollo del proyecto y las soluciones implementadas.
- Comenzar el desarrollo de una aplicación móvil desde la base.
- Diseño de una interfaz intuitiva y accesible para todos los públicos.

3.4 OBJETIVOS PERSONALES

Los objetivos personales de este proyecto han sido en todo momento profesionalizar las aptitudes aprendidas durante la asignatura de Programación Orientada a Objetos y en Desarrollo de Aplicaciones Móviles, extendiendo a su vez mi conocimiento con nuevas herramientas y conceptos como:

- Aprender un uso más avanzado del lenguaje de programación Kotlin para posteriores proyectos/trabajos.
- Realizar un proyecto completo, desde su fase inicial hasta su implementación final, adquiriendo así experiencia de gran utilidad para el futuro.
- Aplicar patrones de diseño y técnicas de programación avanzadas empeladas en el ámbito profesional.

3.5 NO OBEJTIVOS

También se han establecido una seria de no objetivos los cuales no se han querido llevar a cabo por no querer comercializar la aplicación de momento.

Estos no objetivos son:

- No es objetivo desarrollar una aplicación totalmente acabada y lista para su funcionamiento, pero si una primera versión o prototipo.
- Desarrollar la aplicación para IOS [24]. No se desea avanzar en el desarrollo de dicha aplicación en SWIFT [25] para IOS.
- Realizar compraventas y subida de NFTS reales, simplemente poniendo prototipos o copias meramente para su uso didáctico.
- Realizar varias versiones del proyecto para una futura venta.



CAPÍTULO 4

HIPÓTESIS DE TRABAJO

Una vez planteados los objetivos del proyecto, debemos mencionar las herramientas que se han utilizado para el desarrollo de éste.

Desde el nacimiento del proyecto, se tomó la decisión de emplear Kotlin como lenguaje de programación para la aplicación móvil. Esto es debido a diversos factores que veremos más adelante, pero principalmente a lo ágil que es el desarrollo en comparación con otros lenguajes utilizados en el entorno Android y las funcionalidades que posee este lenguaje.

Con el objetivo de mantener un control de las versiones del programa, ya sea para abordar posibles problemas después de realizar cambios o para revisar implementaciones del anterior código, se empleará Sourcetree. Esta herramienta facilita la carga de cada versión, junto con notas que sirvan de guía, en el repositorio de GitHub.

GitHub sirve como un archivo centralizado que almacena todas las versiones de la aplicación, asegurando la trazabilidad y la gestión efectiva del código fuente.

4.1 ANDROID STUDIO

Android Studio es el entorno de desarrollo oficial para Android y está diseñado para trabajar con Kotlin, lenguaje de programación que he escogido para el desarrollo de la aplicación móvil.

Android Studio tiene una gran facilidad de uso ofreciendo una amplia gama de herramientas muy útiles para el desarrollo de aplicaciones móviles, y ante cualquier dificultad o problema puedo encontrar soluciones rápidamente debido a la documentación que hay de esta en páginas oficiales [26].

Además, Android Studio posee un emulador Android integrado el cual se puede ejecutar para ir viendo los resultados de la aplicación en un móvil, probando los diseños y prototipos [26].

Se ha decidido utilizar este entorno debido a que se tenían conocimientos previos con su uso y resulta muy cómodo, poseyendo herramientas prácticas para el desarrollo de Apps.

4.2 KOTLIN

Kotlin es el lenguaje de programación escogido para el desarrollo de esta aplicación móvil. Se trata de un lenguaje de programación moderno y muy adaptable totalmente compatible con Java, poseyendo interoperabilidad, es decir, si tienes un conocimiento de Java puedes emplearlo en Kotlin también [27].

Tiene una sintaxis clara, es fácil de aprender y posee herramientas y características que simplifica el desarrollo de aplicaciones móviles. Tiene una comunidad muy fuerte y se pueden encontrar soluciones a bugs fácilmente [28].

Este lenguaje de programación es el más demandado actualmente en el ámbito profesional de desarrollo de aplicaciones móviles, suponiendo así un reto favorable de aprendizaje para mi futuro.



Ilustración 12 - Logotipo de Kotlin

4.3 METAMASK

Metamask se trata de la billetera crypto más utilizada en el mercado crypto. Mediante esta billetera el usuario puede registrarse o iniciar sesión directamente con la dirección de su billetera. Automáticamente accederá a su perfil el cual estará enlazado con dicha dirección.

Todas las compraventas realizadas en la aplicación móvil irán enlazadas a dicha dirección, dejando registrado dicho movimiento en la Blockchain.

Esta conectividad ha sido la más compleja de este proyecto, debido a la gran complejidad que supone realizar una conexión directa con direcciones y cuentas reales.



Ilustración 13 - Ilustración del depósito de criptomonedas en la billetera Metamask.

4.4 FIGMA

Una de las partes que más importancia se le ha dado durante el desarrollo del proyecto es conseguir un diseño de aplicación avanzada, con una interfaz agradable, vistosa, pero sobre todo intuitiva y simple, para que cualquier usuario sin conocimiento previo pueda poder navegar por la aplicación con mucha facilidad.

Para ello he decidido realizar toda la parte del diseño por FIGMA, ya que se trata de una herramienta para realizar bocetos muy practica y usada en empresas. Posee una amplia gama de distintos dispositivos donde poder consultar los diseños finales sobre estos. También posee una distribución muy buena entre varias páginas de la aplicación [29].

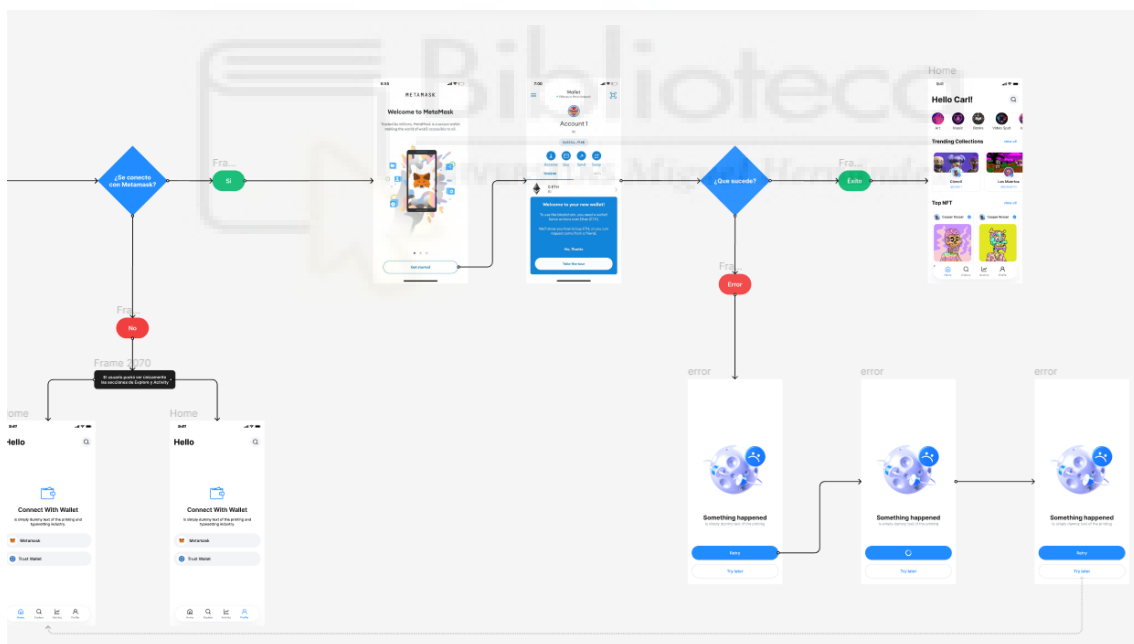


Ilustración 14 - Demostración del flujo de pantallas y su diseño en Figma.

4.5 GIT Y GITHUB - SOURCETREE

Un aspecto muy importante durante el desarrollo de un proyecto de software es el control de versiones. En este campo el rey indiscutible y estándar es GIT. Conjuntamente se ha utilizado los repositorios gratuitos de Github.

Debido a la falta de experiencia y lo solicitado que esta dicha herramienta en el ámbito profesional he decido emplear el control de versiones. Para facilitar la tarea he optado por hacer actualizaciones de versión empleando una interfaz de control de versiones como es Sourcetree [30]

Sourcetree no es más que una herramienta que simplifica la forma en la que el usuario interactúa con los repositorios Git. Esta herramienta permite administrar y visualizar todos los repositorios y versiones con una interfaz muy sencilla [31].



Ilustración 15 - Representación del uso de GitHub y Sourcetree combinados

4.6 CLEAN ARCHITECTURE

La arquitectura de aplicaciones móviles se refiere a la manera en que se construyen y organizan los sistemas, así como a los elementos de diseño que constituyen una aplicación móvil. Incluye las técnicas, procesos y componentes utilizados durante el desarrollo de la aplicación, tomando como punto de partida esos elementos fundamentales.

Crear una arquitectura efectiva para aplicaciones móviles implica una cuidadosa planificación y un diseño estratégico [32].

Seguidamente entraremos en mayor profundidad a conocer el concepto de arquitectura limpia o por capas, como emplearlo en un proyecto y porque es tan importante, y a su vez, las ventajas y desventajas que posee.

4.6.1 DEFINICIÓN

Clean Architecture, también conocida como arquitectura limpia, es un conjunto de prácticas y principios diseñados para facilitar el desarrollo de software mantenible y escalable. Fue propuesta por el desarrollador Robert C. Martin y se basa en la separación de responsabilidades y la independencia de la dirección en capas de una aplicación.

La arquitectura limpia se basa en los siguientes principios [32]:

Independencia de la plataforma: La arquitectura debe ser independiente de la plataforma y no estar vinculada a ningún marco o biblioteca específicos.

Independencia de la interfaz de usuario: La arquitectura debe ser independiente de la interfaz de usuario y no estar vinculada a ninguna tecnología de presentación específica.

Independencia de la base de datos: La arquitectura debe ser independiente de la base de datos y no estar vinculada a ninguna tecnología de almacenamiento específica.

Independencia de cualquier agente externo: La arquitectura debe ser independiente de cualquier agente externo, como el sistema de archivos o la red.

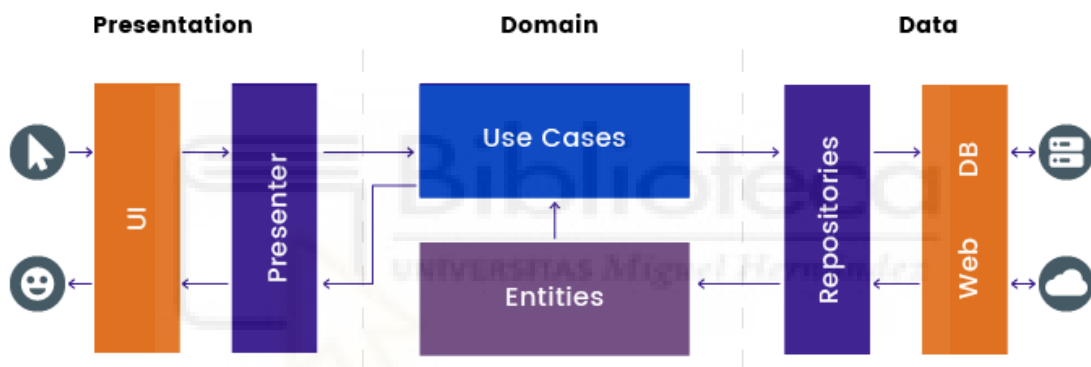


Ilustración 16 - Esqueleto y flujo de una arquitectura limpia.

4.6.2 ESTRUCTURA

Clean Architecture propone organizar el código en capas, cada una con responsabilidades bien definidas. Estas capas, desde la capa más interna hasta la más externa, son [33]:

Entidades: Contiene las entidades de negocio o modelos de datos que representan el núcleo de la aplicación junto con los objetivos. Estas entidades son independientes de cualquier detalle de implementación y lógica externa, no están acopladas a ninguna otra capa encapsulando así los datos principales.

Casos de uso (Use Cases o Interactors): Esta capa contiene la lógica de la aplicación y coordina la interacción entre las entidades del dominio. Aquí se definen los casos de uso de la aplicación, que son las operaciones específicas que puede realizar la aplicación. Estos casos de uso orquestan las acciones de las entidades para lograr los objetivos de la aplicación.

Interfaz de usuario - Adapters (Controladores y Presentadores): Esta capa maneja la interacción del usuario y la presentación de la información conteniendo los adaptadores que son las interfaces con la capa exterior convirtiendo dichos datos en un formato entendible para las capas internas y viceversa. Además, es responsable de traducir las acciones del usuario en invocaciones a los casos de uso.

Frameworks y Drivers: Es la capa más externa de la arquitectura. Aquí se encuentran las implementaciones de detalles externos, como las bases de datos, la interfaz de usuario y otros servicios externos. En esta capa es donde van todos los detalles.

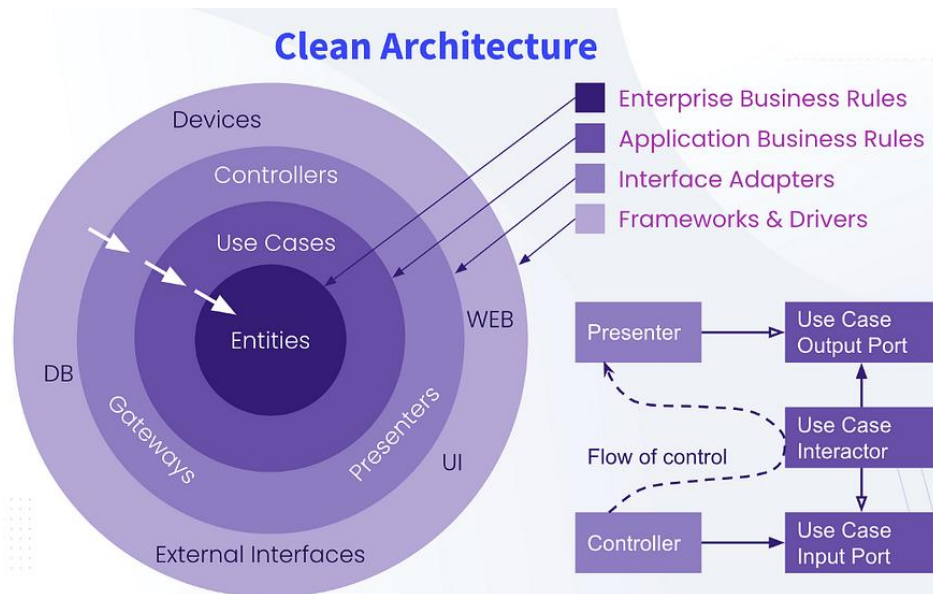


Ilustración 17 - Representación de las capas que componen Clean Architecture con sus componentes y relaciones.

Como se observa en la imagen, este concepto está centrado en un término de encapsulación, donde estas capas solamente pueden apuntar hacia adentro.

Por lo que significa que ninguna capa interna puede conocer información de la capa externa, pero si viceversa. Esta es la principal regla que hace que la arquitectura funcione correctamente [34].

La arquitectura apuntada quedaría así:

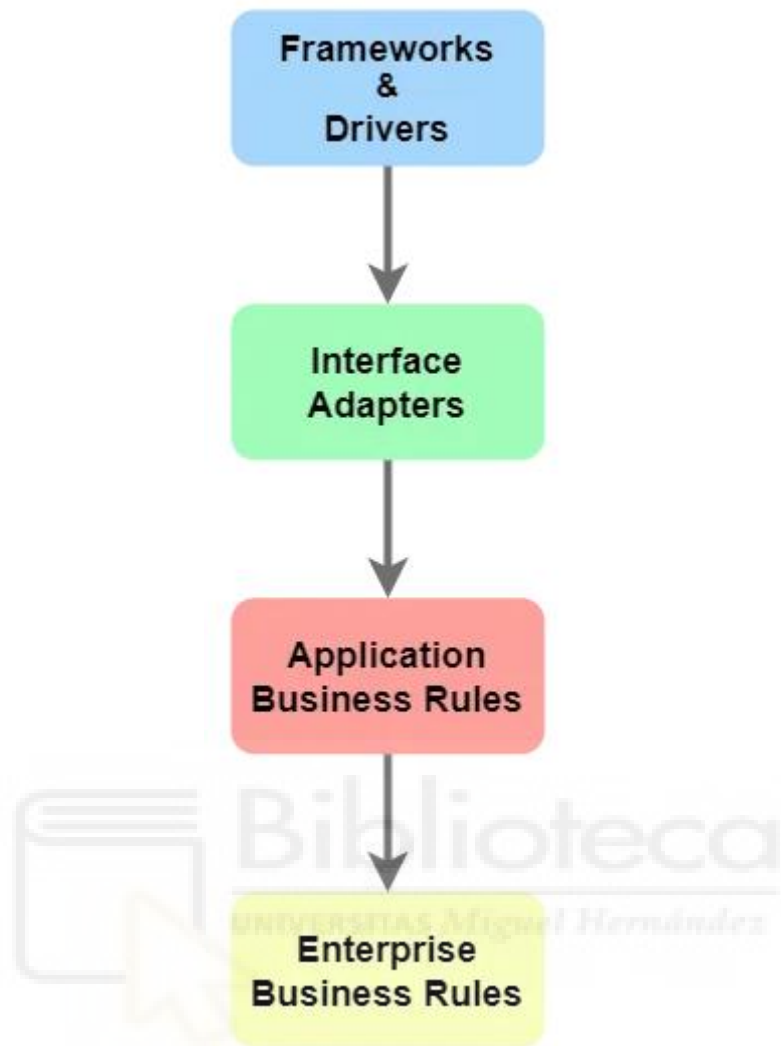


Ilustración 18 - Ilustración que representa la arquitectura simplificada de Clean Architecture.

4.6.3 VENTAJAS

Esta arquitectura se utiliza mucho en el ámbito profesional por varias razones:

Independencia: Cada capa contiene su propio paradigma y no afecta al resto de niveles.

Mantenibilidad y Estructuración: Al organizar el código en capas y separar las preocupaciones se facilita la mantenibilidad del software. Los cambios en una capa no deberían afectar a las demás capas si se respeta la arquitectura y se facilita la búsqueda de funcionalidades.

Escalabilidad: Es posible cambiar o mejorar un componente sin afectar a los demás, lo que facilita la evolución del sistema, pudiendo ser reutilizado.

Facilidad de testeo: Se pueden realizar testeos a cada una de estas capas además de una integración entre sí, sin llegar a afectar a los demás niveles.

Flexibilidad Tecnológica: La arquitectura permite cambiar detalles de implementación, como la base de datos o el framework de interfaz de usuario, sin afectar la lógica central de la aplicación.

En resumen, busca crear sistemas robustos y sencillos de mantener al dividir el código en capas con responsabilidades específicas y minimizar las dependencias entre estas capas. Esta modularidad y separación de preocupaciones son especialmente valiosas en el desarrollo de aplicaciones móviles, donde la evolución rápida y la diversidad de plataformas son comunes [35].

4.6.4 DESVENTAJAS

Realmente esta metodología no presenta ninguna desventaja o complicación más allá de que todo el equipo de desarrollo debe conocer la metodología que se aplica siendo responsables de entender y aplicar estas reglas.

En este caso, al realizar el proyecto yo solo ha sido más sencillo llevar estas reglas a cabo desde un comienzo, pero la velocidad de desarrollo del proyecto disminuye considerablemente debido a la estructuración previa [35]



4.7 MVVM

Si se desea conseguir una aplicación móvil escalable mantenible y fácil de testear, la combinación de MVVM [36] con Clean Architecture es una buena práctica ya que van ligados y se completan a la perfección.

MVVM incluyendo una arquitectura limpia es una técnica muy profesional y avanzada ideal en proyectos grandes con la que se consigue ir un paso más allá de la separación de responsabilidades dentro de la aplicación, consiguiendo la abstracción de la lógica de las acciones

4.7.1 DEFINICIÓN

MVVM, que significa Modelo-Vista-ViewModel, se trata de un patrón de arquitectura de software empleado en el desarrollo de aplicaciones Android. El objetivo principal es realizar una separación de la interfaz de usuario (View) de la parte lógica (Model), consiguiendo así un aspecto visual completamente independiente [37].

Si tenemos en cuenta este modelo, esta sería una breve descripción de cada componente de MVVM:

Modelo (Model): Representa los datos y la lógica de “negocio” de la aplicación. Es la capa donde se establecen los modelos para el almacenamiento de datos.

Vista (View): Contiene la interfaz de usuario de la aplicación. Es responsable de mostrar los datos al usuario y de enviar las acciones del usuario al View Model. Si nos centramos en contexto Android, se implementan como una actividad o Fragment [38].

View Model: Actúa como intermediario entre el modelo y la vista, contiene la lógica de presentación siendo responsable de preparar los datos para que sean mostrados en la vista. Maneja las interacciones del usuario y actualiza el modelo en consecuencia, estando diseñado para ser independiente de la interfaz de usuario.

Por lo que el flujo de datos quedaría así:

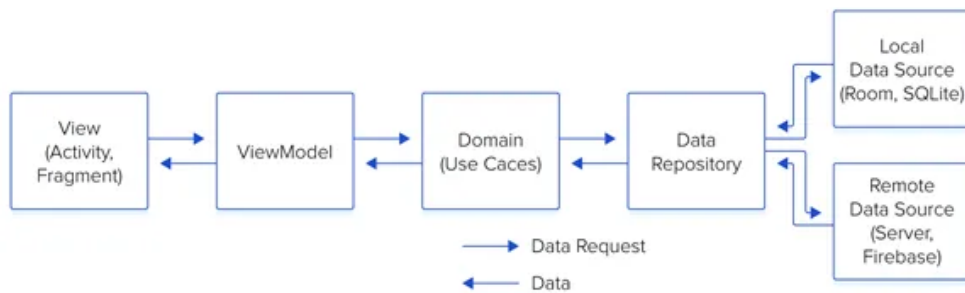


Ilustración 19 - Esqueleto y flujo de MVVM. Direcciones de Data y Data Request.



4.7.2 IMPLEMENTACIÓN

MVVM, destaca por su capacidad de separar las preocupaciones de una aplicación, facilitando la prueba unitaria de cada componente de manera aislada.

Facilita la gestión del ciclo de vida de la interfaz de usuario en entornos como Android, donde las actividades y fragmentos pueden ser destruidos y recreados en respuesta a diversos eventos. [39]

En Android, el patrón MVVM se emplea junto con bibliotecas como Android ViewModel y LiveData, las cuales han sido empleadas en este proyecto, las cuales facilitan mucho la implementación y la observación de datos en tiempo real. Estas bibliotecas proporcionan un conjunto de componentes y pautas cuya finalidad es conseguir una aplicación Android moderna y robusta [40].

En esta imagen podemos consultar el resultado de combinar una Clean Architecture con el modelo MVVM:

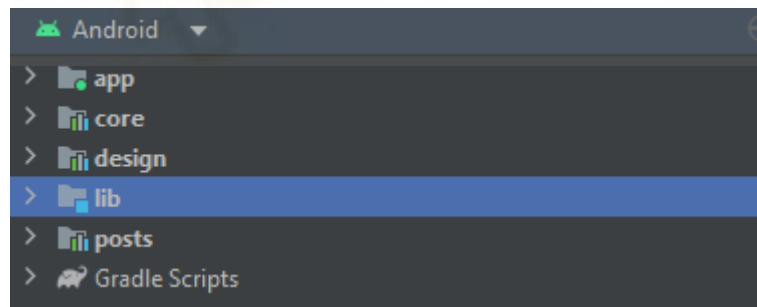


Ilustración 20 - Organización paquetes del proyecto.

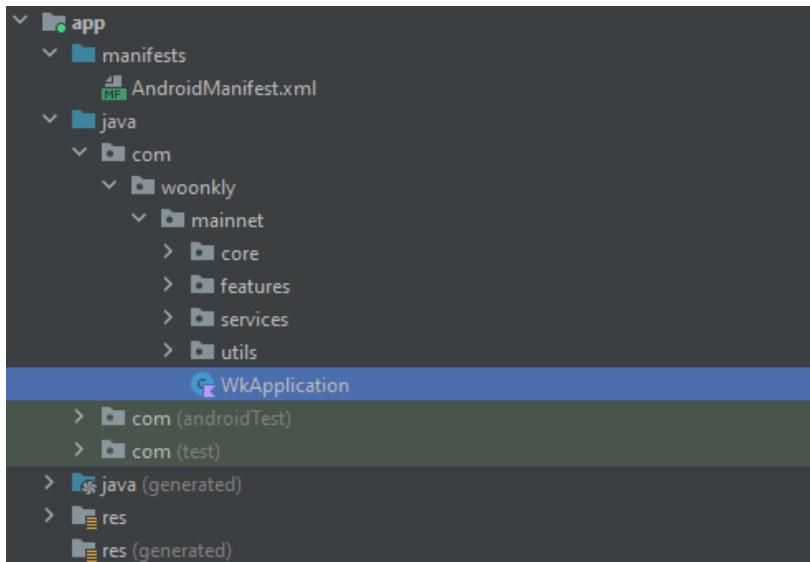


Ilustración 21 - Representación división de paquetes internos del proyecto (Core, Features, Services, Utils).

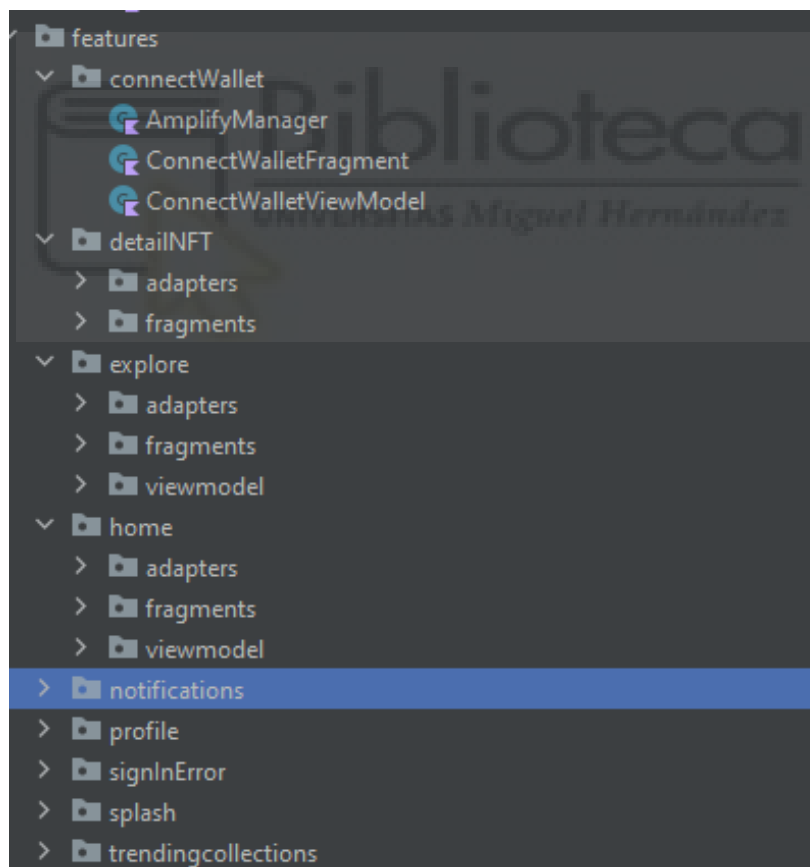


Ilustración 22 - División de los paquetes por cada pantalla y sus funciones.

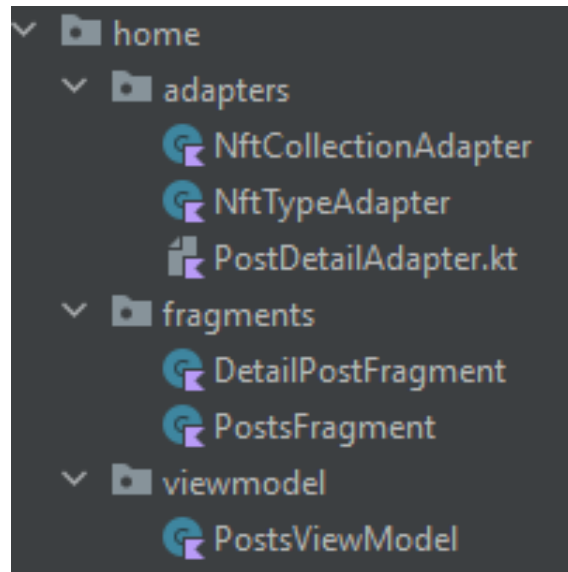


Ilustración 23 - Representación de división por capas de MVVM

En esta última imagen se puede observar la división entre adapters-fragments-viewmodel

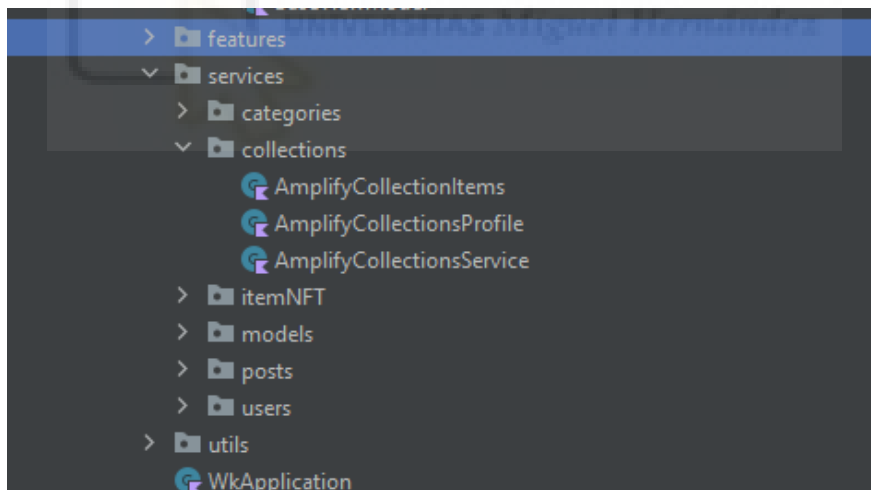


Ilustración 24 - Contenido de los feautres y las clases para llamadas al servidor.

Y en esta ultima la división de las features, de los servicios que contienen las llamadas al servidor, y de las utilidades de la aplicación.

4.8 DEPENDENCIAS

En el desarrollo de este proyecto necesitas diversas bibliotecas y herramientas que desempeñan un papel fundamental en la funcionalidad y rendimiento de la aplicación.

Estas dependencias que se utilizan para agregar las funcionalidades adicionales que posee su código [41].

En este proyecto han hecho faltas diversas librerías externas para conseguir el resultado final deseado, empleando cada una para un determinado fin. Las dependencias usadas han sido:

- **Clouinary**
- **Android X**
- **AWS Amplify Android**
- **GSON**

Estas dependencias se han de añadir en el apartado Gradle [42] y hay que seguir una metodología estricta para que funcionen correctamente, la cual se ha conseguido mediante investigación y tutoriales.

Así se vería estas dependencias en el archivo Gradle:

```
dependencies {
    implementation project(':core')
    testImplementation project(':core')
    implementation project(':posts')
    implementation project(':design')
    //Navigation
    implementation 'androidx.navigation:navigation-fragment-ktx:2.4.1'
    implementation 'androidx.navigation:navigation-ui-ktx:2.4.1'
    //AWS
    implementation 'com.amplifyframework:aws-api:1.35.3'
    implementation 'com.amplifyframework:aws-datastore:1.35.3'
    implementation ('com.amplifyframework:aws-auth-cognito:1.35.3')
    implementation 'com.amazonaws:aws-android-sdk-apigateway-core:2.52.1'
    implementation 'com.amazonaws:aws-android-sdk-mobile-client:2.52.1'
    implementation 'com.amazonaws:aws-android-sdk-auth-userpools:2.52.1'
    implementation 'com.amplifyframework:core-kotlin:0.21.5'

    //Cloudinary
    implementation 'com.cloudinary:kotlin-url-gen:1.0.0'

    // Wallet Connect
    implementation project(":lib")
    implementation "com.github.komputing.khex:extensions:1.1.0"
    implementation "org.java-websocket:Java-WebSocket:1.4.0"
    implementation 'com.squareup.moshi:moshi:1.8.0'
    implementation "com.squareup.okhttp3:okhttp:4.9.1"

    //GSON
    implementation 'com.google.code.gson:gson:2.8.5'
```

Ilustración 25 - Archivo Gradle del proyecto.

4.8.1 CLOUDINARY

Cloudinary es una plataforma de gestión de medios en la nube, la cual ofrece un almacenamiento, manipulación y entrega de imágenes o videos.

La integración de Cloudinary es muy sencilla y se realiza mediante su correspondiente SDK, la cual facilita la comunicación efectiva con la plataforma desde la aplicación Android [43].



Ilustración 26 - Logotipo de Cloudinary.

4.8.2 ANDROID X

AndroidX es un conjunto de bibliotecas y extensiones de Android que proporcionan funciones y características adicionales para el desarrollo de aplicaciones Android modernas. Al migrar a AndroidX, se obtiene el acceso a las últimas API de Android, correcciones de errores y mejoras de rendimiento.

Esta dependencia es esencial para garantizar la compatibilidad con las versiones más recientes de Android y aprovechar las últimas características disponibles en la plataforma [44].

4.8.3 AWS AMPLIFY ANDROID

AWS Amplify Android es una biblioteca que simplifica la integración de servicios de Amazon Web Services en aplicaciones Android. Amplify ofrece una forma fácil de configurar y utilizar servicios en la nube, así como almacenamiento, autenticación y bases de datos proporcionando así la posibilidad de crear aplicaciones escalables y seguras [45].

Amplify Android es la forma recomendada para crear aplicaciones nativas de Android con AWS.

En este proyecto ha supuesto la parte más complicada puesto que si no se tiene experiencia previa puede ser muy complejo y tienes que saber cómo la petición al servidor se realiza y que tipo de datos devuelve.

En este caso se ha empleado API (REST) que es una de las interacciones cliente servidor que proporciona esta herramienta. Se trata de una comunicación mediante el protocolo HTTP [46] para obtener datos o ejecutar operaciones los cuales se encuentran en XML o JSON [47]

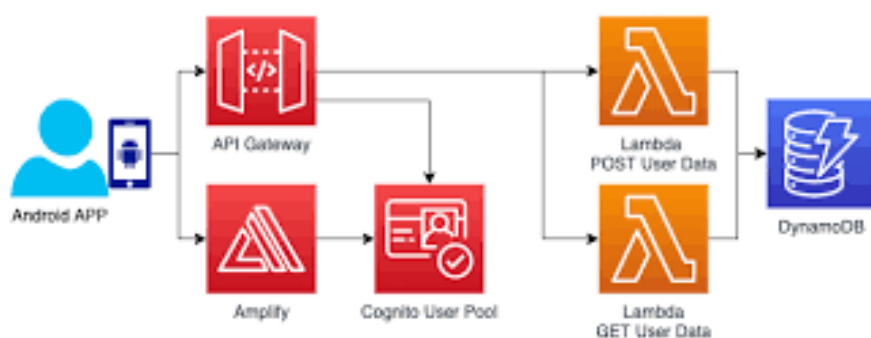


Ilustración 27 Esqueleto y funcionamiento de la API REST.

4.8.4 GSON

En el contexto de esta aplicación GSON se utiliza para manejar la conversión de datos entre el formato JSON que devuelve la API REST comentada anteriormente, a objetos en el código fuente de la aplicación para que estos sean individuales y poder operar sobre estos.

Esta dependencia es crucial en este proyecto para la correcta interpretación y manipulación de datos estructurados en el sistema y ha supuesto además un reto de aprendizaje y manejo de esta técnica de programación.

4.8.5 CONCLUSIÓN

Este apartado proporciona una visión detallada de las dependencias clave utilizadas en el desarrollo de la aplicación, destacando su relevancia en aspectos como la gestión de medios, compatibilidad con Android, integración con servicios en la nube de AWS y manipulación de datos en formato JSON.



CAPÍTULO 5

METODOLOGÍA Y RESULTADOS

En esta sección se va a desarrollar la metodología utilizada para este proyecto, y se justificará por qué se han empleado ciertas técnicas y no otras.

5.1 CICLO DE VIDA DEL PROYECTO

El ciclo de vida de un proyecto de software contempla las fases necesarias para validar el desarrollo del software y así garantizar que este cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo, asegurándose de que los métodos usados son apropiados.

Su origen radica en que es muy costoso rectificar los posibles errores que se detectan tarde en la fase de implementación. Utilizando metodologías apropiadas, se podría detectar a tiempo para que los programadores puedan centrarse en la calidad del software, cumpliendo los plazos y los costes asociados [48].

Existen diferentes tipos de ciclo de vida según las características y necesidades del proyecto siendo los más reconocidos y utilizados, que son:

- Ciclo de vida en cascada
- Ciclo de vida iterativo
- Ciclo de vida en espiral
- Ciclo de vida ágil.

Este proyecto es muy extenso en cuanto a diseño y estructuración correcta de la aplicación, pero la idea principal está muy bien definida desde el principio de este. Esta aplicación no necesitará grandes cambios durante su desarrollo, ya que parte de un diseño amigable predefinido y la estructura y código de la aplicación móvil seguirá los principios de MVVM y Clean Architecture.

Por este motivo, se ha elegido el modelo de ciclo de vida en cascada mejorado, un modelo con éxito demostrado en numerosos proyectos de desarrollo de software.

Este tipo de ciclo de vida es muy útil para proyectos bien definidos desde su comienzo, con todos los objetivos a cumplir bien definidos y una estructura de aplicación muy clara y que no necesita muchos cambios [49].

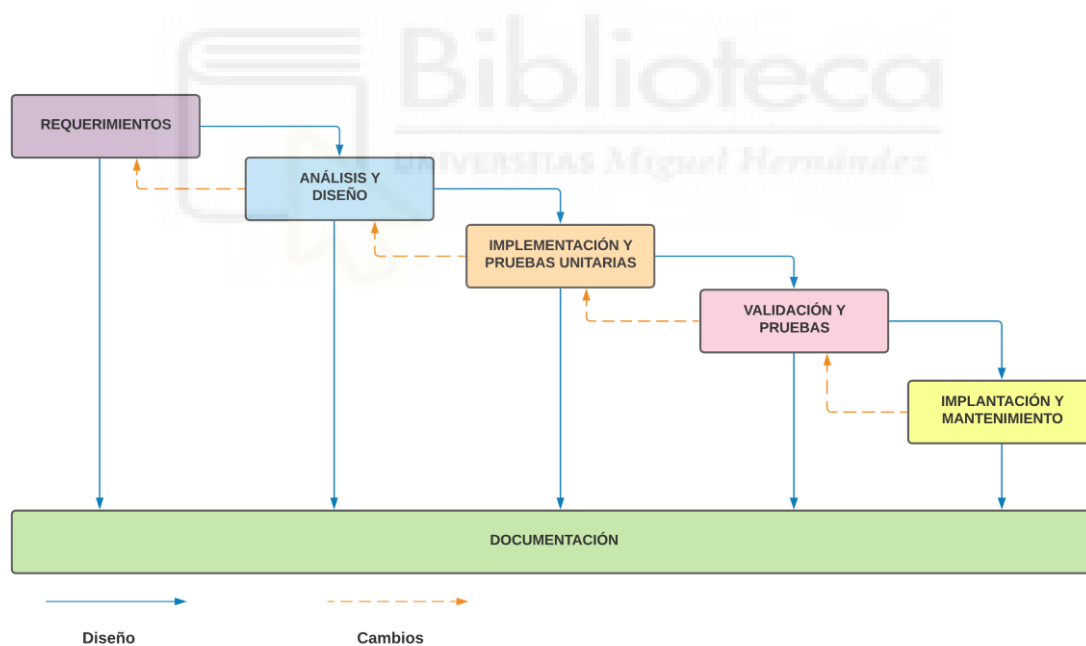


Ilustración 28 - Ciclo de vida en cascada seguido.

Éste ha sido el ciclo de cascada empleado durante el proyecto, diferenciando las fases de desarrollo.

5.1.1 REQUERIMIENTOS

Los requerimientos definen las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. Los requerimientos sirven como brújula que guiará el desarrollo del software, definiendo el comportamiento esperado, las necesidades empresariales y la interacción con los usuarios.

Por dicho motivo, en esta fase se ha de hacer una recopilación de los problemas y dificultades que encuentran los usuarios en otros mercados NFTs para poder resolverlos y mejorarlos, documentando estos para poder realizar una interfaz ideal para todos los públicos respaldando nuestras conclusiones.

El proyecto no solo aspira a solucionar estos problemas, si no ha diseñar una interfaz ideal para todos los públicos, considerando la diversidad de usuarios

Es crucial establecer con claridad los objetivos fundamentales del proyecto desde el principio, ya que actúan como criterios de validación, asegurando que cada requisito abordado satisfaga las necesidades y expectativas [50].

Dentro de la parte más técnica y de desarrollo de software se centra en un desarrollo que cumpla ciertas características ya comentadas, Clean Architecture y MVVM.

5.1.2 ANÁLISIS Y DISEÑO

Este apartado comprende el análisis de los requerimientos y el diseño de los componentes de la aplicación referentes al código, interfaz de usuario, bases de datos, etc. [50].

En primer lugar, tras realizar el estudio comentado anteriormente de los problemas del usuario y de la estructuración de otros mercados NFTs, el primer paso es realizar un análisis y el diseño de la interfaz de la aplicación.

5.1.2.1 DIAGRAMA DE GANTT

Durante el desarrollo de proyecto es importante mantener una planificación de todo el proceso seguido a la hora de realizar el proyecto.

Es por eso por lo que se ha establecido un diagrama de Gantt en este proyecto, enfocado en dividir las principales tareas y subtareas que se han ido realizando durante el desarrollo de la aplicación móvil.

Nombre de tarea	Asignado	Fecha de inicio	Fecha final	Estado
		2023/09/18	2023/12/14	
1 Planificación del Proyecto	S	2023/09/18	2023/09/18	Terminado
2 Investigación del proyecto	S	2023/09/25	2023/09/25	Terminado
3 <input type="checkbox"/> Diseño Interfaz Aplicación		2023/09/25	2023/10/13	
3.1 Diseño Interfaz de Pantallas	S	2023/09/25	2023/09/28	Terminado
3.2 Diseño de elementos de cada pantalla	S	2023/10/02	2023/10/05	Terminado
3.3 Navegación de la Aplicación	S	2023/10/06	2023/10/10	Terminado
3.4 Estudio de Interfaz	S	2023/10/11	2023/10/13	Terminado
+ Añadir una tarea Añadir un hito				
4 <input type="checkbox"/> Desarrollo de la Aplicación		2023/10/16	2023/12/12	
4.1 Esqueleto de la aplicación	S	2023/10/16	2023/10/19	Terminado
4.2 Estructuración de paquetes	S	2023/10/19	2023/10/20	Terminado
4.3 Codificación de la Interfaz	S	2023/10/23	2023/10/26	Terminado
4.4 Codificación de las Pantallas	S	2023/10/26	2023/10/27	Terminado
4.5 Investigación conexión Metamask	S	2023/10/30	2023/11/02	Terminado
4.6 Desarrollo conexión Metamask	S	2023/11/03	2023/11/09	Terminado
4.7 Investigación AWS Amplify	S	2023/11/10	2023/11/15	Terminado
4.8 Investigación Cloudinary	S	2023/11/16	2023/11/17	Terminado
4.9 Desarrollo primera llamada al servidor	S	2023/11/20	2023/11/24	Terminado
4.10 Investigación de conversión de JSON a Clase Objeto	S	2023/11/27	2023/11/29	Terminado
4.11 Conversión de Datos JSON a Clase Objeto	S	2023/11/29	2023/11/30	Terminado
4.12 Adecuación de Datos e Interfaz	S	2023/11/30	2023/12/01	Terminado
4.13 Desarrollo llamadas al servidor restantes	S	2023/12/04	2023/12/06	Terminado
4.14 Testeo en Dispositivo Móvil Físico	S	2023/12/08	2023/12/08	Terminado
4.15 Implementación de mejoras	S	2023/12/11	2023/12/12	Terminado
+ Añadir una tarea Añadir un hito				
5 <input type="checkbox"/> Comienzo Memoria Final		2023/10/16	2023/12/12	
5.1 Introducción	S	2023/10/16	2023/10/20	Terminado
5.2 Estado de la cuestión	S	2023/10/23	2023/10/26	Terminado
5.3 Objetivos del proyecto	S	2023/10/26	2023/10/27	Terminado
5.4 Hipótesis de trabajo	S	2023/12/06	2023/12/08	Terminado
5.5 Metodología y Resultados	S	2023/12/08	2023/12/11	En progreso
5.6 Conclusiones	S	2023/12/11	2023/12/12	Abierto
+ Añadir una tarea Añadir un hito				
6 Revisiones finales del proyecto	S	2023/12/13	2023/12/14	Abierto

Ilustración 29 - Tareas diagrama de Gantt.

Así es como quedaría finalmente el diagrama de Gantt completo.

Las principales tareas o hitos en los cuales se ha dividido la planificación del proyecto en el diagrama de Gantt han sido:

- **Planificación del proyecto.**
- **Investigación del proyecto.**
- **Diseño Interfaz de la Aplicación.**

- **Desarrollo de la Aplicación.**
- **Memoria Final.**
- **Revisiones Finales.**

Estas tareas principales a su vez son divididas en otras subtarear que se han llevado a cabo dentro del mismo hito de desarrollo.

El proyecto comienza con la misma planificación de éste, en el cual se quiere establecer hacia donde vamos a conducir el proyecto, que se desea hacer y si está dentro de nuestras posibilidades.

Esta tarea va seguida inmediatamente de una investigación de la propia idea de proyecto, en la cual se realiza una investigación más profunda de que abordar en el proyecto y en qué se puede llegar a diferenciar. En este caso, se procede a realizar una investigación de otros mercados de NFTs, la interfaz de usuario que poseen, como están desarrolladas, qué facilidad de uso tienen, etc.

Seguidamente, comienza el diseño de la aplicación. Esta parte del proyecto trata de plasmar en la interfaz de la aplicación la facilidad de uso y la usabilidad amigable, accesible para todos los públicos, siendo esta distintiva y una versión mejorada de otros mercados de NFTs. En esta etapa se procede a diseñar en Figma las pantallas de la aplicación y las vistas detalladas de los elementos que posee la aplicación.

Una vez finalizada la etapa de diseño, se procede a codificar esta interfaz previamente establecida, implantando toda la parte técnica y funcional de la aplicación. Se comienza a codificar la estructura y división de la aplicación, seguidamente cada pantalla de la aplicación y sus funciones relacionadas.

Esta parte del proyecto es la más costosa en cuanto a tiempo empleado y dificultad de desarrollo, debido a que en algunas fases se establece un tiempo de

investigación del funcionamiento de ciertas implementaciones en el código. Al ser un lenguaje de programación el cual no se ha estudiado previamente, le da un plus de dificultad.

Finalmente, una vez el desarrollo de la aplicación haya llegado a su fin se procede a ir finalizando la memoria final, en la que ciertas partes se han ido realizando durante el transcurso de partes previas del diagrama de Gantt, como puede ser la introducción, el estado de la cuestión y los objetivos del proyecto.

Poco antes de terminar el proyecto y finalizar la memoria se procede a hacer una revisión más exhaustiva de que todo este realizado de manera correcta y finalizada.

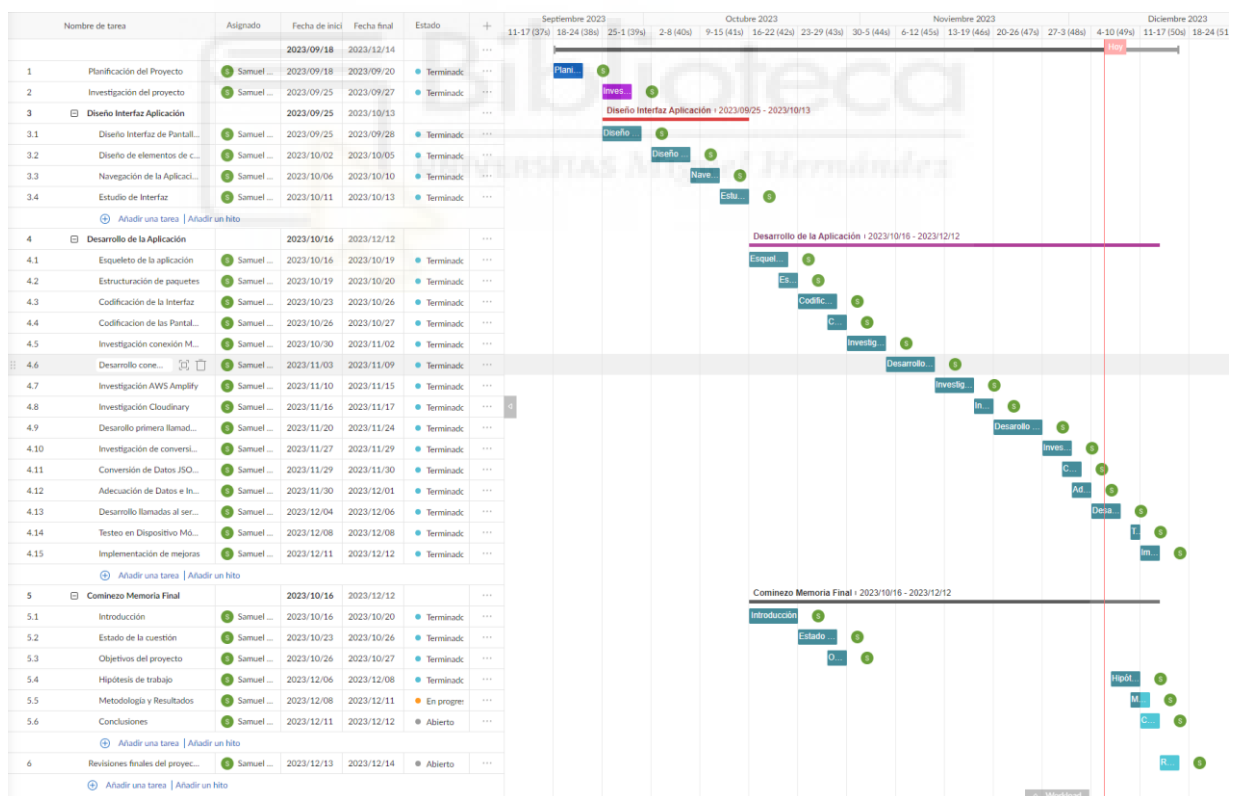


Ilustración 30 - Vista en semanas del diagrama de Grant.

Como se puede apreciar en esta imagen [30], se establecen las tareas con una fecha de inicio y una fecha de fin ideal, para mantener una buena dinámica y poder finalizar el proyecto dentro del plazo asignado.

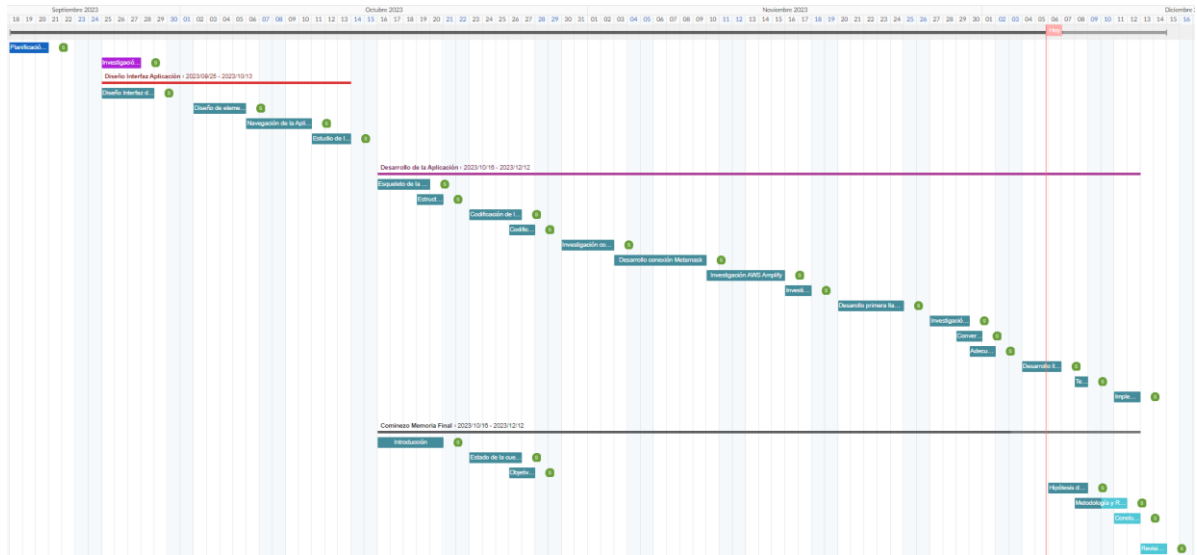


Ilustración 31 - Ilustración en cascada del Diagrama de Grant

Cabe destacar que, el ciclo de vida del proyecto se realiza en cascada como previamente se ha explicado, durante el cual se han podido implementar ciertos retoques donde se hacían correcciones o mejoras en cada sección.

5.1.2.2 ACTORES Y CASOS DE USO

En el ámbito del diseño y análisis de sistemas, los casos de uso y los actores desempeñan un papel fundamental al modelar la interacción entre el usuario y el sistema.

Un caso de uso se define como una secuencia estructurada de acciones que el sistema lleva a cabo en respuesta a eventos iniciados por un actor. Cada caso de uso presenta al menos un escenario, proporcionando una visión detallada de cómo debería desarrollarse la interacción entre el sistema y el usuario.

Por otro lado, los actores, representan entidades externas al sistema que participan en esta interacción. Cada actor actúa como un elemento clave al modelar las diferentes formas en el que el sistema puede ser utilizado. En esencia, los actores son la representación conceptual de cualquier cosa o cualquier persona que interactúe con el sistema, facilitando así una comprensión clara y estructurada de las interacciones entre el usuario y la funcionalidad del sistema [52].

A continuación, se especifican 2 partes diferenciales y que están relacionadas entre sí, los actores y los casos de uso que forman parte del sistema. Para ello, es necesario determinar una plantilla para los actores y otra para los casos de uso.

Las plantillas utilizadas para definir a los actores o los casos de uso son las siguientes:

Actor	<Id>	<Nombre descriptivo>
Descripción	<Rol del actor en el sistema>	
Comentarios	<Comentarios adicionales sobre el actor>	
Casos de uso en los que interviene	<Casos de uso en los que el actor tiene participación>	

Caso de uso	<Id>	<Nombre descriptivo>	
Actor principal	<Actor principal>		
Actores secundarios	<Actores secundarios>		
Descripción	<Descripción, lo que hace y no hace, y cómo lo hace>		
Precondición	<Precondición del caso de uso>		
Secuencia normal	Paso	Acción	
	Paso 1	<Secuencia de acciones>	
	Paso 2		
	Paso N		
Postcondición	<Condición posterior que debe darse en este caso de uso>		
Excepciones	Paso	Acción	
	Paso 1	<Secuencia de acciones en caso de error>	
	Paso 2		
	Paso N		
Comentarios	<Comentarios adicionales sobre el caso de uso>		

5.1.2.2.1 ACTORES

Actor	A-01	Usuario
Descripción	Usuario externo que quiere realizar uso de la aplicación.	
Comentarios	Requiere descargar la aplicación previamente.	
Casos de uso en los que interviene	<ul style="list-style-type: none"> - C-01: Iniciar sesión - C-02: Editar perfil - C-03: Consultar NFTs a la venta - C-04: Consultar las colecciones de NFTs que posee el usuario - C-05: Consultar los NFTs que posee el usuario. - C-06: Consultar listado "Trending Collections" - C-07: Consultar listado "Top NFTs" - C-08: Ver detalles de un NFT - C-09: Ver detalles de una colección 	

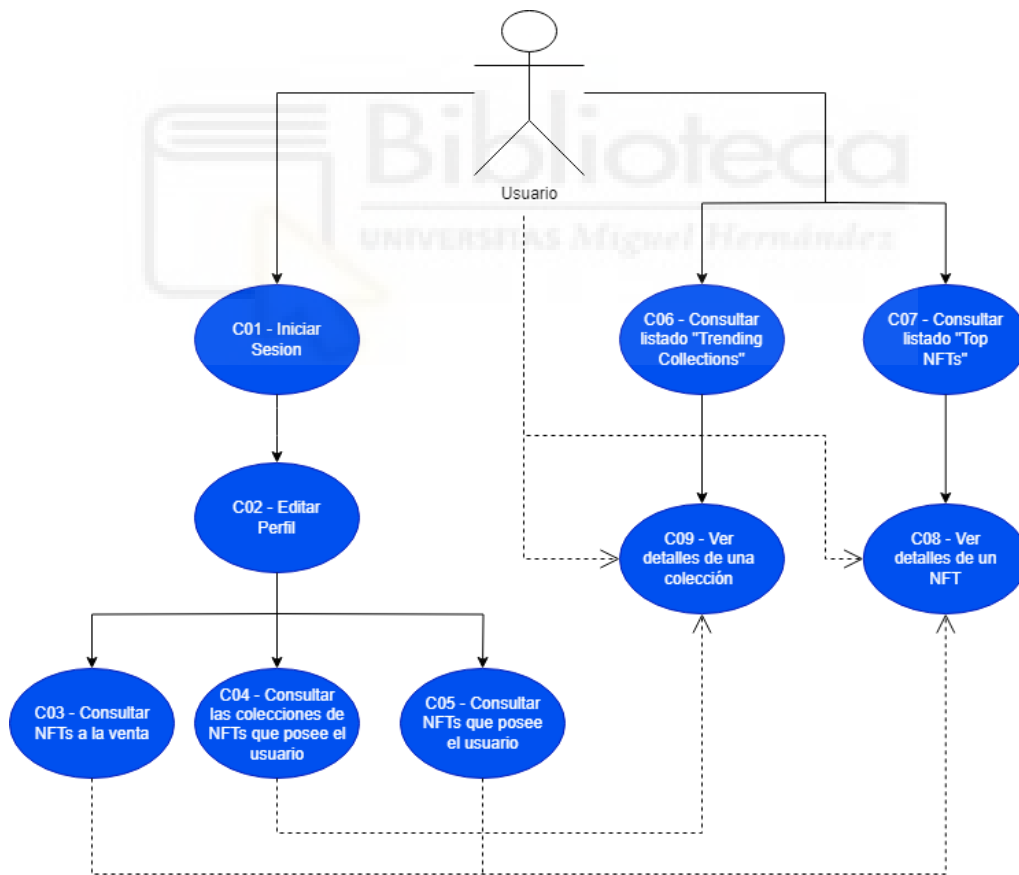


Ilustración 32 - Casos de uso del usuario

5.1.2.2.2 CASOS DE USO

Caso de uso	C-01	Iniciar de sesión	
Actor principal	Usuario		
Actores secundarios	-		
Descripción	Inicio de sesión a través de Metamask, incluyendo su dirección Blockchain.		
Precondición	Poseer cuenta en Metamask y una billetera cripto valida.		
Secuencia normal	Paso	Acción	
	Paso 1	El usuario abre la aplicación.	
	Paso 2	El usuario salta el mensaje de bienvenida.	
	Paso 3	El usuario selecciona el inicio de sesión mediante Metamask.	
	Paso 4	Se redirige a la aplicación Metamask.	
	Paso 5	Creación de contrato Blockchain.	
	Paso 6	Firma contrato Blockchain.	
	Paso 7	Aceptación del contrato.	
	Paso 8	Enlace dirección billetera Metamask.	
	Paso 9	El sistema automáticamente redirige a la aplicación, manteniendo la sesión iniciada con la dirección de la billetera.	
	Paso 10	Carga automática de todo el contenido de la aplicación (NFTs, colecciones, usuarios, etc)	
Postcondición	El usuario quedara registrado automáticamente mediante su dirección Blockchain. El resto de los campos de usuario quedaran guardados por defecto y el usuario podrá editarlos.		
Excepciones	Paso	Acción	
	Paso 4	El usuario no tiene la aplicación descargada en su dispositivo móvil. El usuario deberá descargarla y volver a comenzar el proceso.	
	Paso 7	Ha ocurrido algún error de conexión o creación de la firma. Se redirige al usuario a una pantalla nueva, donde muestra el error, permite volverlo a intentar o volver atrás.	
Comentarios	-		

Caso de uso	C-02	Editar perfil	
Actor principal	Usuario		
Actores secundarios	-		
Descripción	Edición de campos que componen el perfil del usuario.		
Precondición	Haber realizado un previo inicio de sesión.		
Secuencia normal	Paso	Acción	
	Paso 1	El usuario inicia sesión.	
	Paso 2	Acceder a la sección de perfil.	
	Paso 3	Clicar el botón de 3 puntitos.	
	Paso 4	Seleccionar la opción editar perfil del desplegable.	
	Paso 5	Se accede a la pantalla de edición.	
	Paso 6	Seleccionar campo que se desea editar.	
	Paso 7	Editar el campo con el nuevo contenido.	
	Paso 8	Guardar cambios.	
Postcondición	El usuario no puede editar el campo de dirección de la billetera, va asociado a metamask. Los datos se actualizan correctamente y se guardaran junto con la billetera del usuario.		
Excepciones	Paso	Acción	
	Paso 4	El usuario desea editar el campo de dirección de billetera, se informará de que no es posible.	
	Paso 8	No se guardan o actualizan los cambios. Se mostrará una alerta indicando dicho error.	
Comentarios	-		

C-03	Consultar NFTs a la venta.	
Actor principal	Usuario	
Actores secundarios	-	
Descripción	Visualizar en el perfil los NFTs que posee el usuario y están a la venta.	
Precondición	Haber realizado un previo inicio de sesión.	
Secuencia normal	Paso	Acción
	Paso 1	El usuario inicia sesión.
	Paso 2	Acceder a la sección de perfil.
	Paso 3	Se desplaza o hace clic en la opción "On Sale".
	Paso 4	Se realiza una llamada al servidor, solicitando los NFTs asociados a la billetera del usuario, con el filtro de que estén en venta.
	Paso 5	Se cargan los NFTs que posee el usuario en venta.
Postcondición	Si el usuario no posee NFTs o no tiene ninguno puesto en venta, la lista de NFTs aparecerá vacía.	
Excepciones	Paso	Acción
Comentarios	-	

C-04	Consultar las colecciones de NFTs que posee el usuario.	
Actor principal	Usuario	
Actores secundarios	-	
Descripción	Visualizar en el perfil las colecciones a las que pertenecen los NFTs del usuario.	
Precondición	Haber realizado un previo inicio de sesión.	
Secuencia normal	Paso	Acción
	Paso 1	El usuario inicia sesión.
	Paso 2	Acceder a la sección de perfil.
	Paso 3	Se desplaza o hace clic en la opción "Collections".
	Paso 4	Se realiza una llamada al servidor, solicitando los NFTs asociados a la billetera del usuario.
	Paso 5	Se cargan las colecciones a las que pertenecen esos NFTs
Postcondición	Si el usuario no posee NFTs o no tiene ninguno puesto en venta, la lista de colecciones aparecerá vacía.	
Excepciones	Paso	Acción

Comentarios	Se mostrará una pequeña previsualización y resumen de dichas colecciones.
--------------------	---

C-05	Consultar NFTs que posee el usuario.	
Actor principal	Usuario	
Actores secundarios	-	
Descripción	Visualizar en el perfil los NFTs que posee el usuario.	
Precondición	Haber realizado un previo inicio de sesión.	
Secuencia normal	Paso	Acción
	Paso 1	El usuario inicia sesión.
	Paso 2	Acceder a la sección de perfil.
	Paso 3	Se desplaza o hace clic en la opción "Collectibles".
	Paso 4	Se realiza una llamada al servidor, solicitando los NFTs asociados a la billetera del usuario.
	Paso 5	Se cargan los NFTs que posee el usuario.
Postcondición	Si el usuario no posee NFTs, la lista aparecerá vacía,	
Excepciones	Paso	Acción
Comentarios	-	

C-06	Consultar listado "Trending Collections"	
Actor principal	Usuario	
Actores secundarios	-	
Descripción	Visualizar un listado de las colecciones de moda del momento.	
Precondición	-	
Secuencia normal	Paso	Acción
	Paso 1	Acceder a la aplicación.
	Paso 2	Acceder a la sección de home.
	Paso 3	Realizar click en el botón "view all" en la previsualización del listado de las colecciones.
	Paso 4	Se accede a otra pantalla donde se muestra una lista de colecciones de moda.
	Paso 5	Se cargan los contenidos de estas colecciones.
Postcondición	-	
Excepciones	Paso	Acción

Comentarios	-
--------------------	---

C-07	Consultar listado “TOP NFTs”	
Actor principal	Usuario	
Actores secundarios	-	
Descripción	Visualizar un listado de los NFTs más top del momento.	
Precondición	-	
Secuencia normal	Paso	Acción
	Paso 1	Acceder a la aplicación.
	Paso 2	Acceder a la sección de home.
	Paso 3	Realizar click en el botón “view all” en la previsualización del listado de los top NFTs.
	Paso 4	Se accede a otra pantalla donde se muestra una lista de los mejores o más destacados NFTs.
	Paso 5	Se cargan los contenidos de estos de NFTs.
Postcondición	-	
Excepciones	Paso	Acción
Comentarios	-	

C-08	Ver detalles de un NFT.	
Actor principal	Usuario	
Actores secundarios	-	
Descripción	El usuario selecciona un NFT y visualiza detalles adicionales.	
Precondición	-	
Secuencia normal	Paso	Acción
	Paso 1	El usuario accede a la aplicación.
	Paso 2	Realiza clic en un NFT.
	Paso 3	Se accede a otra pantalla donde se muestran los detalles del NFT.
	Paso 4	Se muestra el usuario que posee el NFT actualmente, el nombre del NFT, una breve descripción, el número de la colección que es, y el precio actual.
	Paso 5	Se abre un desplegable abajo donde se puede consultar los usuarios que han realizado transacciones con dicho NFT y los últimos movimientos de este NTF (compraventas).
Postcondición	-	
Excepciones	Paso	Acción
Comentarios	-	

C-09	Ver detalles de una colección.	
Actor principal	Usuario	
Actores secundarios	-	
Descripción	El usuario selecciona una colección y visualiza detalles adicionales.	
Precondición	-	
Secuencia normal	Paso	Acción
	Paso 1	El usuario accede a la aplicación.
	Paso 2	Realiza clic en una colección.
	Paso 3	Se accede a otra pantalla donde se muestran los detalles de la colección.
	Paso 4	Se muestra el usuario que creó la colección, el nombre de la colección, las fotos, una breve descripción, el número de seguidores, los

		ítems, owners, el volumen de tardes y el floor Price,
	Paso 5	Se abre un desplegable abajo donde se puede consultar los NFTs que forman parte de esa colección.
Postcondición	-	
Excepciones	Paso	Acción
Comentarios	-	



5.1.2.3 DIAGRAMAS DE ACTIVIDAD

Los diagramas de actividad constituyen una herramienta esencial en la Ingeniería del Software para la representación visual del comportamiento interno y dinámico del sistema, en este caso, la propia aplicación.

Estos diagramas ofrecen una perspectiva de alto nivel del flujo de programas permitiendo a los usuarios externos al proceso de desarrollo, comprender con mayor claridad las condiciones y limitaciones asociadas a operaciones específicas, así como visualizar de manera intuitiva el flujo entre diversas actividades [51].

A continuación, se incluyen los diferentes diagramas de la actividad de la aplicación desarrollada.



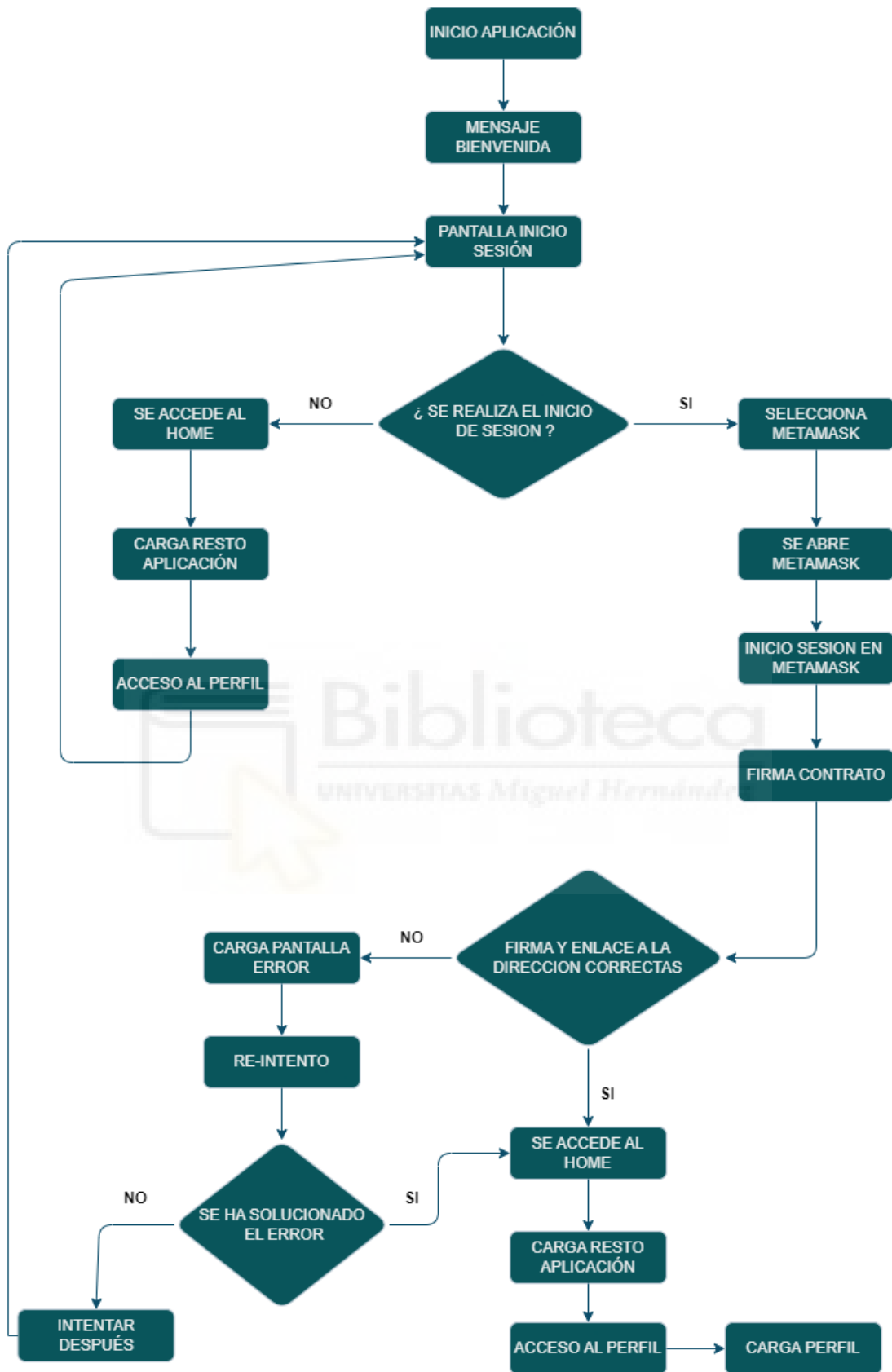


Ilustración 33 - Ilustración Diagrama de actividad inicio de aplicación

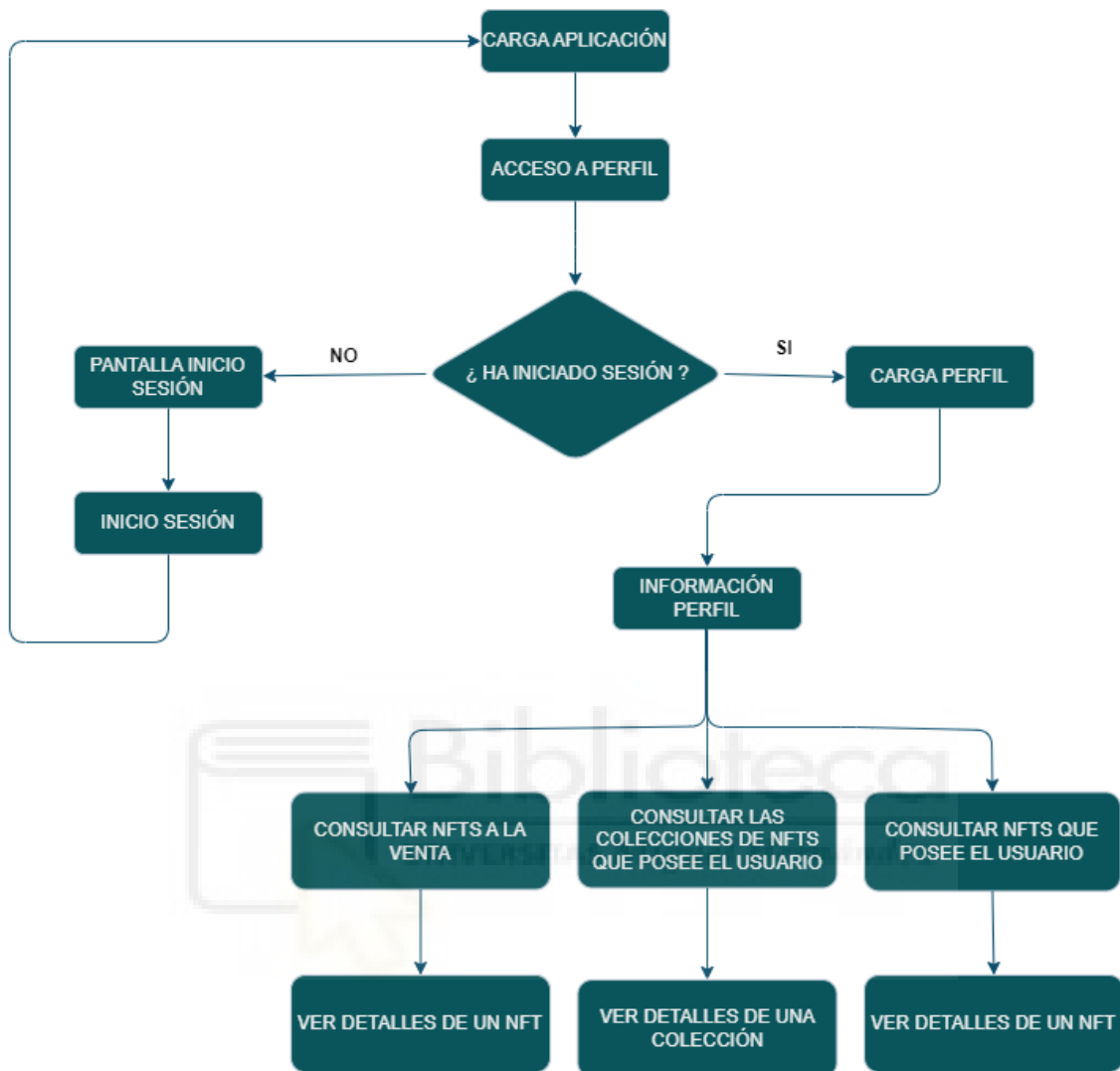


Ilustración 34 - Ilustración Diagrama de actividad consulta información perfil.



Ilustración 35 - Ilustración Diagrama de actividad consulta del Home

5.1.2.4 DISEÑO INTERFAZ

Mediante Figma, se establece los diagramas de flujo y la división de la aplicación en distintas pantallas y secciones, obteniendo así la arquitectura general del sistema.

En primer lugar, se establece como será el flujo de datos para el inicio de sesión, que como hemos comentado anteriormente, será mientras Metamask, la cual se puede observar con este diseño:

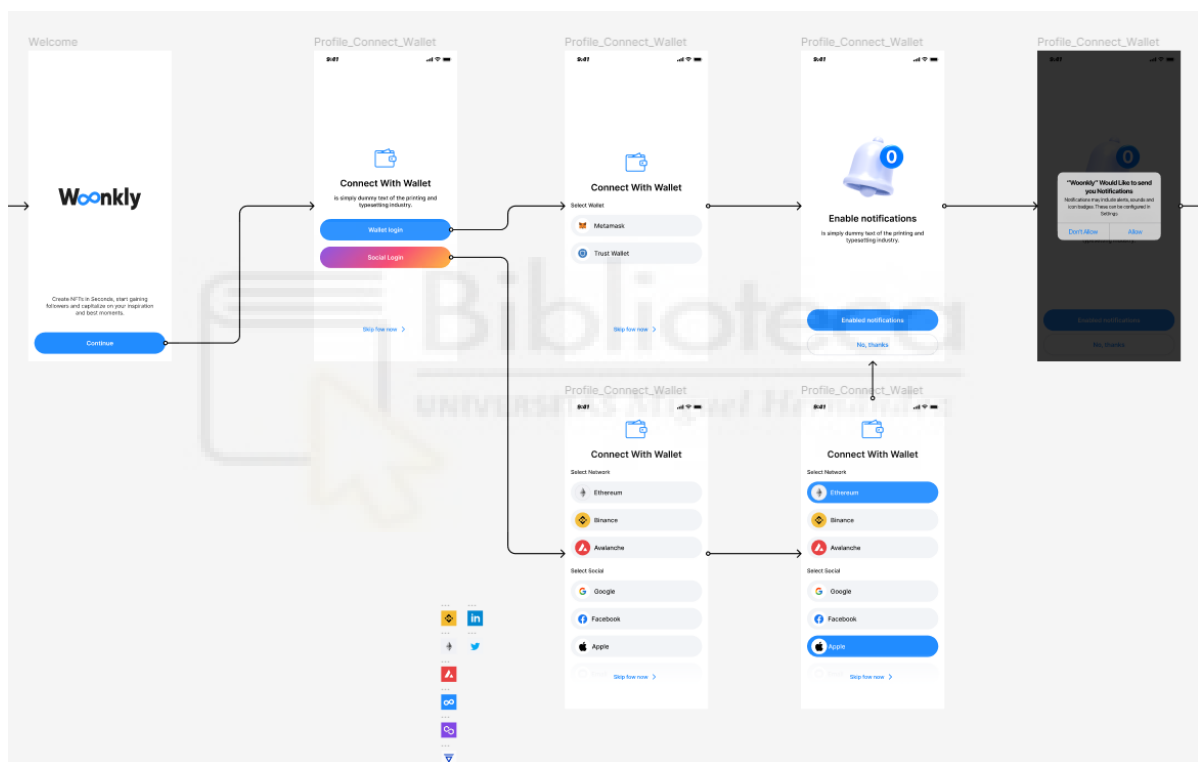


Ilustración 36 - Inicio sesión en la aplicación

El usuario podrá iniciar sesión con una billetera cripto, o saltar este paso. Una vez claro cómo se realizará el inicio de sesión se establece como estará dividida la aplicación y que sectores de navegación tendrá y cuál será su funcionalidad. Dentro de la aplicación una vez iniciada la sesión, se distribuye en Home, Feed, Explore, y Profile.

Si nos centramos en la navegación total de la aplicación podemos consultar internamente en el código el nav_graph, el cual contiene toda la navegación de acciones de la aplicación móvil.

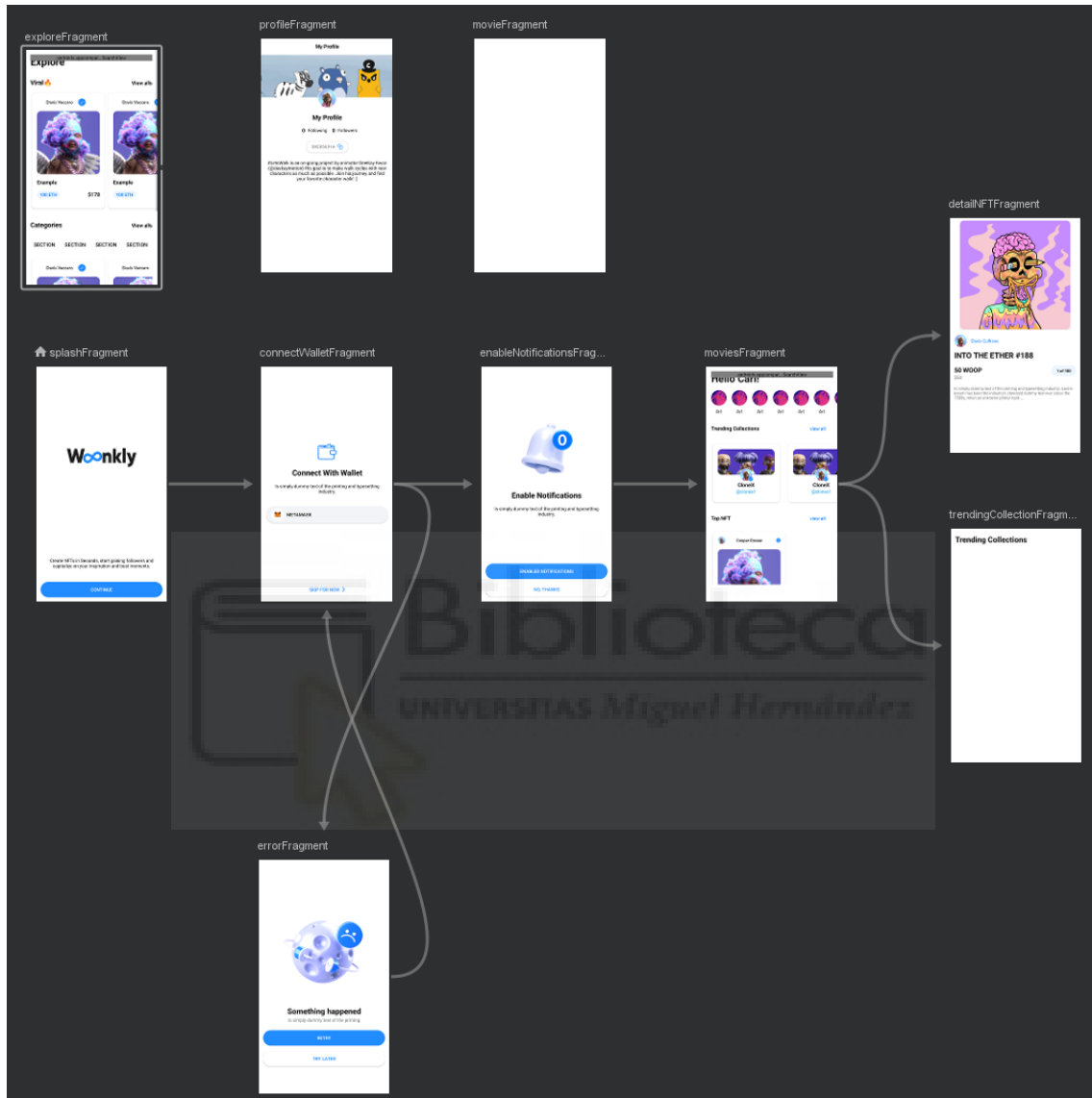


Ilustración 37 - Navegación de acciones internas de la aplicación.

5.1.2.4.1 HOME

Esta sección es a la que se accede una vez iniciada la sesión o se salta ese paso. Contiene todo el contenido funcional de la aplicación a modo resumen. En la sección superior aparecen todas las categorías disponibles de NFTs, en las cuales se encontrarán las colecciones. Seguidamente, más abajo aparecen un breve resumen de unas 5 colecciones de NFTs que están de moda y a continuación, se muestran los NFTs con más renombre. Se pueden explorar más variedad de secciones accediendo al botón "view all", el cual abrirá un desplegable de búsqueda.

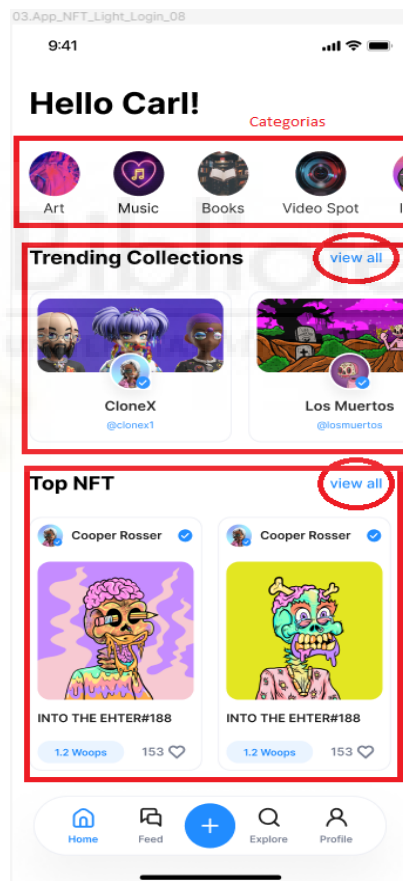


Ilustración 38 -Home de la aplicación.

5.1.2.4.2 EXPLORE

Esta sección es donde se puede explorar todo el mercado de NFTs disponibles. Se puede consultar Colecciones, NFTs, Usuarios, Rangos raros de NFTs. Este apartado está pensado para que tengas todo en uno de una manera muy sencilla de encontrar y navegar por ella, de tal forma que se muestre una interfaz amigable e intuitiva.

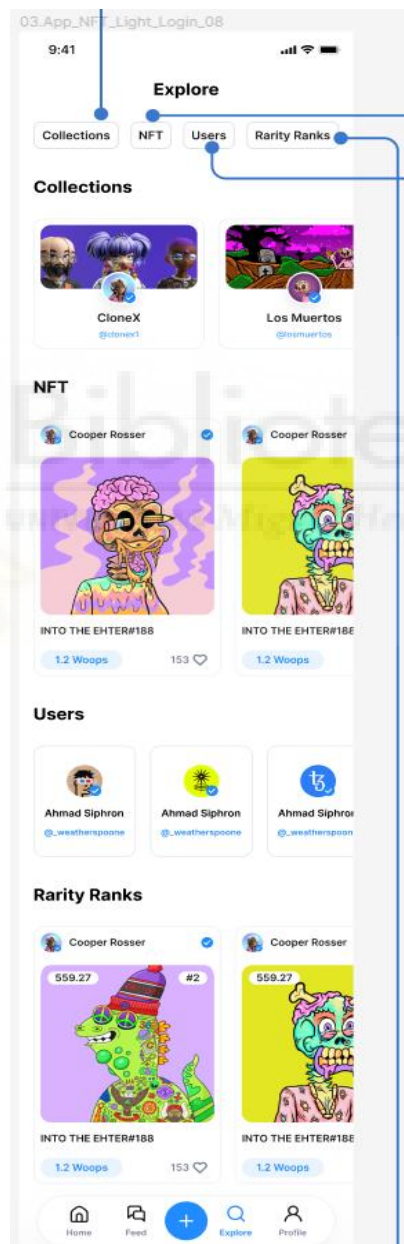


Ilustración 39 - Sección Explore de la aplicación.

A modo resumen, aparece un listado de unos 5/10 elementos de cada sección, en modo previsualización, para que el usuario pueda hacerse una idea de qué aparecerá y como aparecerá.

Cuando se accede a una opción de exploración de la parte superior, se ocultan las demás secciones y se amplían el número de elementos mostrados.

A continuación, se muestra un ejemplo con usuarios:

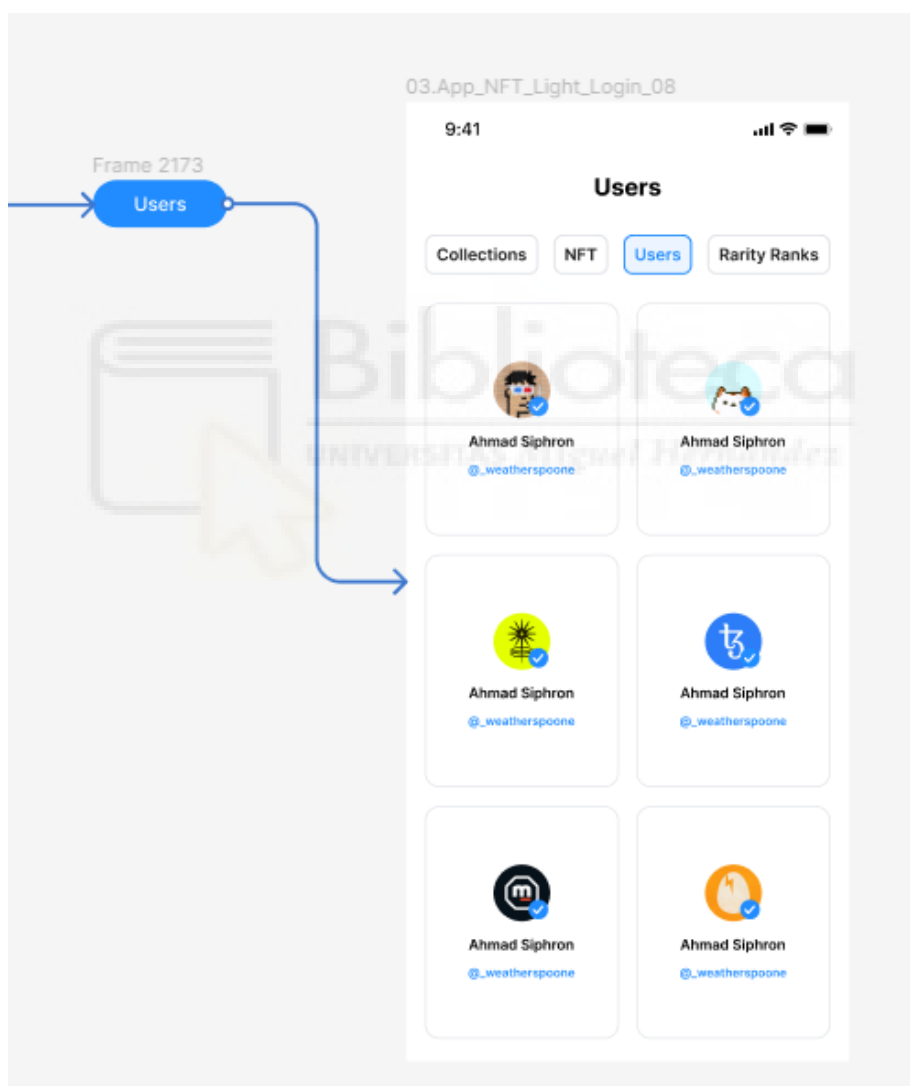


Ilustración 40 - Listado usuarios en el apartado explore de la aplicación.

5.1.2.4.3 PROFILE

Se trata del apartado en el que el usuario puede acceder a su perfil, tras haber iniciado sesión. Si no había iniciado sesión se le redirige otra vez a la pantalla de inicio de sesión.

En este apartado se mostrará la dirección asociada a dicho perfil, con el cual todas las transacciones que se realicen con dicho perfil quedarán registradas y vinculadas con esa dirección de la billetera (registro en la Blockchain).

El apartado contiene una foto de perfil, un banner o foto de fondo, el nombre del usuario, el número de personas que sigue y le siguen, la dirección de su billetera y una breve descripción. Estos datos son editables en cualquier momento.

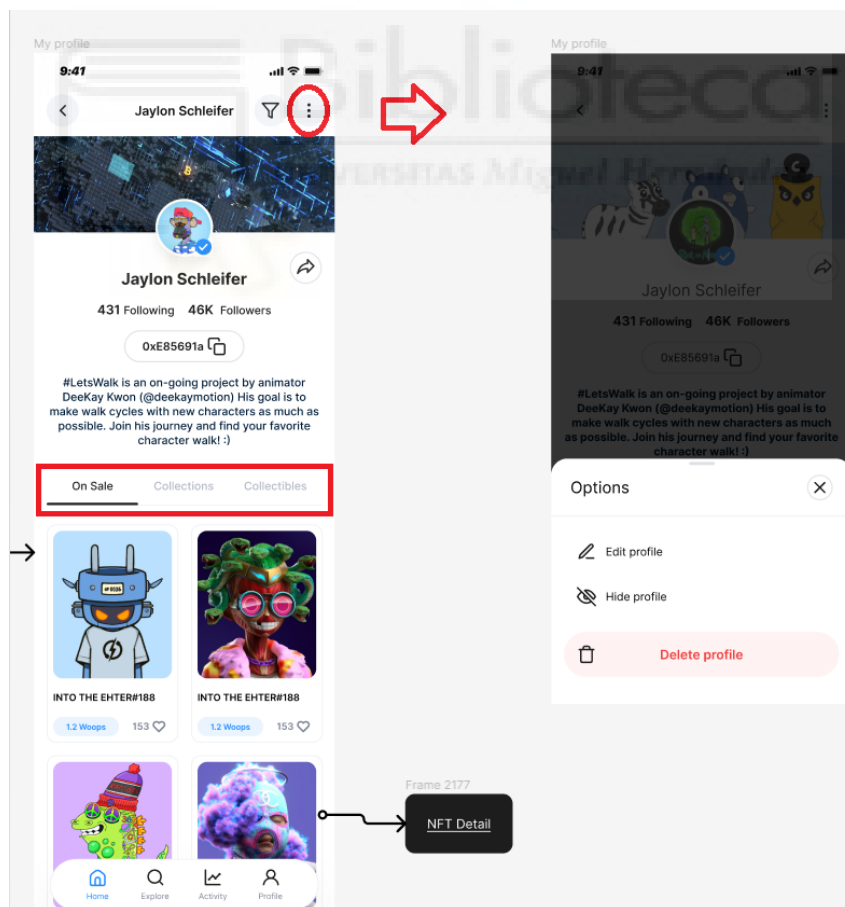


Ilustración 41 - Perfil de usuario en la aplicación

A su vez este apartado contiene 3 secciones distintas desde las que el usuario puede consultar qué NFTs posee el usuario y tiene en venta, las colecciones a las que pertenecen los NFTs que posea, para poder obtener una vista general de dicha colección, y en collectibles, los NFTs que tiene asociados a dicho perfil/dirección.

5.1.2.4.4 BOTON +

En este apartado los usuarios/artistas podrán subir un NFT propio con las licencias adecuadas y firmando contratos de comisiones, autenticidad y seguridad. Del mismo modo, pueden subir una colección de NFTs y dar de alta esta colección en el mercado de NFTs como nueva y comerciable.

Es importante destacar que el desarrollo y diseño final de esta sección aún no se ha llevado a cabo, y se ha dejado para una fase posterior de implementación de esta, por lo que se incluye en la sección de desarrollos futuros.

5.1.2.4.5 FEED

Esta sección de la aplicación está diseñada para que los usuarios compartan sus comentarios sobre la aplicación, proporcionen sugerencias de mejoras y reporten posibles problemas, incluyendo la identificación de colecciones fraudulentas. Además, podrán expresar sus opiniones sobre NFTs y agregar comentarios diversos.

Este apartado se presenta en forma de blog, permitiendo a cada usuario proporcionar feedback de manera interactiva.

Como en el caso del "Boton +", su desarrollo e implementación se ha dejado para una versión posterior de la aplicación.

5.1.3 IMPLEMENTACIÓN Y PRUEBAS UNITARIAS

Esta fase del ciclo consiste en traducir el diseño a código, siguiendo las mejores prácticas de desarrollo. En esta parte se realiza la búsqueda de errores, mejoras y se realizan varias pruebas.

Se van desarrollando distintos componentes por separado y se comprueban a través de dichas pruebas unitarias, para ver su correcto funcionamiento, integrándolas finalmente tras su éxito en la aplicación [50].

Como se ha comentado con anterioridad, la aplicación se codificará utilizando Kotlin, a través de Android Studio. La aplicación se comunicará mediante consultas REST con el servidor para obtener la información necesaria de la base de datos (AWS AMPLIFY y Cloudinary).

Cada nuevo elemento deberá ser probado para comprobar su correcto funcionamiento. Tras finalizar una prueba con éxito y considerando que el diseño y funcionamiento de dicho elemento es el deseado, se procede a añadir dicho elemento al producto final comprobando una correcta estructuración.

5.1.3.1 PRUEBAS Y ERRORES

Durante el proceso de pruebas se han ido obteniendo errores los cuales han sido ciertamente complejos de solucionar. Estos errores implicaban una gran cantidad de tiempo, debido a la falta de conocimientos avanzados en el desarrollo de

aplicaciones en Kotlin, lo que retrasó encontrar exactamente una resolución a dichos problemas.

A continuación, se detallan algunos errores principales que se produjeron durante el proceso de desarrollo:

Conexión Metamask: Durante las pruebas de conexión con una billetera real se han tenido varios problemas que se han tenido que ir abordando. El primer gran problema fue que la conexión de Metamask real no se puede realizar desde el emulador que ofrece Android Studio, se necesita la aplicación oficial para los contratos de conexión y la firma de autenticidad y de consentimiento.

Para solucionar este problema se realiza la conexión a un dispositivo móvil físico con la aplicación descargada y con una billetera cripto real (no es necesario poseer criptomonedas) en la cual va a ser ejecutada la aplicación.

Se van realizando pruebas con el dispositivo móvil hasta que se consigue exitosamente el inicio de sesión mediante Metamask, obteniendo el contrato de seguridad firmado y aceptado por el usuario.

Llamada a Amplify: Éste ha sido el mayor de los problemas a abordar. Al no estar familiarizado con el uso de APIS en el entorno Android esta tarea ha sido la más costosa.

El principal problema era la codificación de esta llamada, ya que resultó ser más costosa de lo esperado y necesitaba un formato muy concreto, además de tener que dar de alta variables del entorno y dentro del propio proyecto, en el cual se da de alta el servidor al que se le hacen las peticiones REST.

Tras varios tutoriales y con la ayuda de un programador senior que sí que había trabajado con Amplify, se consiguió configurar adecuadamente las peticiones al servidor, obteniendo la respuesta en forma de datos.

Transformación de JSON a una clase Objeto: Otro problema abordado durante el desarrollo de esta aplicación fue el tratamiento de los datos que devolvía la llamada a Amplify.

Mediante esta petición de API REST se consigue una respuesta por parte del servidor, donde se enviaban datos en formato JSON.

Para el tratamiento de estos datos lo ideal era que cada dato interno y numerado de JSON fuera tratado como un objeto. Es decir, cuando se realizaba una llamada al servidor para realizar la carga de las colecciones de NFTs de moda, se devolvían dichos NFTs numerados en formato JSON. Pero se recibían unos datos que globalizaba todos los NFTs de manera conjunta, sin ningún divisor, y no se conseguía realizar un tratamiento individual de estos.

Tras varias consultas se logró realizar una función que realizaba la transformación de estos datos a una clase objeto, por lo que se podía tratar cada NFT de forma individual, como objeto propio.

Se utilizó un conversor de JSON a data class Object para poder ver mejor como se distribuye.

5.1.4 VALIDACIÓN Y PRUEBAS

Esta etapa del ciclo trata de la verificación de que el software desarrollado realiza correctamente las tareas indicadas en la especificación con un diseño adecuado y listo para su testeo [50].

En la fase previa ya se ha comprobado que los distintos elementos de la aplicación funcionan bien por separado, ahora se debe comprobar que todo el conjunto de elementos funciona de forma adecuada.

Si el diseño es el correcto y la funcionalidad mediante código no genera errores se procede a su testeo global.

Si se producen errores o bugs, se corrigen para que la aplicación final no contenga estos errores, y los elementos de la aplicación funcionen en su totalidad, realizando testeos reales con el emulador de Android o un dispositivo móvil físico, accediendo a las peticiones por servidor y comprobando si la carga global de estas es correcta.

Si todas estas pruebas se han finalizado correctamente, se procede a realizar una actualización de versiones. Es aquí donde entran en juego GitHub y Sourcetree.

Una vez todas las pruebas son exitosas y se ha llegado al punto final de funcionamiento deseado, se realiza una actualización en el repositorio, a través de Sourcetree, añadiendo una nota explicativa de que se ha implementado o mejorado/arreglado.

5.1.5 IMPLANTACIÓN Y MATENIMIENTO

En esta fase final del ciclo, es en la que el sistema es completamente funcional y preparado para ser usado. Es en este momento donde los usuarios de la aplicación toman protagonismo y entran a escena, haciendo uso de la aplicación sin conocer los detalles técnicos de la misma.

Contar con la ayuda de mis padres ha sido fundamental, para que, sin tener conocimiento previo y sin entender los mercados de NFTs, fueran capaces de poder navegar en la aplicación sin aparente dificultad. Su valioso feedback, enfocado en la facilidad o dificultad percibida al encontrar ciertos elementos, contribuyó positivamente al proceso.

Surgieron ciertos problemas que no fueron detectados durante el proceso de desarrollo, como la falta de información para iniciar sesión con Metamask, y la falta de una versión en el idioma Castellano, ya que solo se ha desarrollado la aplicación enfocada en Ingles. Estos cambios fueron tomados en cuenta para futuras versiones.

Con un número de usuarios superior a los que se han involucrado en las fases de prueba anteriores, y considerando su falta de familiaridad con el funcionamiento interno de la aplicación, es lógico que puedan surgir nuevos errores o mejoras posibles. Una vez realizado los testeos con usuarios finales, la aplicación estará lista para su mantenimiento y futuras mejoras.



CAPÍTULO 6

CONCLUSIONES Y PROPUESTAS

6.1 CONCLUSIONES

A nivel personal, este proyecto ha supuesto varios retos. En primer lugar, abordar el aprendizaje de un nuevo lenguaje de programación como Kotlin y la familiarización con herramientas externas como AWS Amplify y Cloudinary, adoptando una metodología profesional. Por otro lado, la dificultad que supone comenzar un proyecto desde 0 de manera personal, siendo muy complejo el proceso de inicialización del proyecto, dándole forma a los diseños pensados y su transformación a código.

La experiencia de trabajar con Kotlin de la mano de Android Studio ha sido muy positiva. Al compartir similitudes con Java, un lenguaje que ya había sido empleado en asignaturas previas del grado, y pudiendo ser implementado en Kotlin, se partía con una base de conocimientos previos. Esta familiaridad, sumada al deseo de especializarme en el desarrollo de aplicaciones móviles, ha convertido este desafío en un logro exitoso.

Desde hace un par de años, también se despertó en mi un cierto interés en la tecnología Blockchain y en el descubrimiento de nuevas formas de inversión. En dichos años se han experimentado el uso de diversas aplicaciones de billeteras, de exchanges, y de mercados de NFTs. Durante este uso, se ha nutrido la idea de que las interfaces de usuario de estas aplicaciones no son lo suficientemente usables y adaptables para cualquier público, provocando que, personas sin

ciertos conocimientos o usos de aplicaciones parecidas, tengan problemas de usabilidad.

Es por ese motivo por el cual se propuso realizar este proyecto, combinando la realización de una aplicación con un diseño usable e intuitivo, disponible para todos los públicos, y las ganas de realizar una formación profesional para optar a futuros puestos de trabajo en el ámbito de desarrollo de aplicaciones móviles.

En términos generales, se ha conseguido elaborar un proyecto desde cero, utilizando tecnologías y técnicas aprendidas durante el Grado de Ingeniería Informática, como Ingeniería del Software, Gestión de proyectos de Ingeniería del Software, Interfaces de Usuario, y Desarrollo de Aplicaciones para Dispositivos Móviles, para su análisis y desarrollo, combinándolo a su vez con el aprendizaje de nuevas herramientas que no habían sido utilizadas durante la carrera.



6.2 TRABAJOS FUTUROS

En lo que respecta al futuro para este proyecto, no está pensado para su comercialización, pero puede ser usado a modo de presentación en el portfolio, mostrando el proyecto realizado por mí a las empresas que presenten ofertas laborales. Otra alternativa, también sería contactar con un mercado de NFTs de ámbito regional, que no disponga de aplicación móvil, brindándoles la oportunidad de expandir su presencia en el mercado y marca personal.

Desde la perspectiva técnica, hay áreas clave que merecen atención y desarrollo adicional. Entre ellas destaca la implementación del diseño y desarrollo del botón de subida de NFTs y de la sección de Feed. También se podría incluir otras formas de inicio de sesión, que no sean Metamask, y se podría realizar más pruebas con usuarios finales para obtener un feedback valioso y considerar posibles mejoras.

Adicionalmente, se plantea la posibilidad de mejorar la velocidad de respuesta en las llamadas a las API. Aunque hasta ahora se han probado de manera satisfactoria con una cantidad de datos manejable, es esencial llevar a cabo pruebas de rendimiento más exhaustivas al incrementar considerablemente la carga de datos. Este enfoque garantizará un desempeño óptimo a medida que el proyecto evolucione y se expanda.

En resumen, el proyecto presenta un potencial considerable tanto como una herramienta para la presentación profesional como para la colaboración estratégica con mercados de NFTS en crecimiento.



CAPÍTULO 7

BIBLIOGRAFÍA

7.1 FUENTES BIBLIOGRÁFICAS

- [1] NFTs - <https://www.xataka.com/basics/que-nft-como-funcionan> - 08/10/2023
- [2] Blockchain - <https://www.blockchain.com/es/> - 08/10/2023
- [3] ¿Qué es un NFT y para qué sirve? - <https://www.softonic.com/articulos/que-es-nft-para-que-sirve> - 09/10/2023
- [4] Bitcoin - <https://bitcoin.org/es/> - 09/10/2023
- [5] Ethereum - <https://ethereum.org/es/> - 09/10/2023
- [6] ¿Qué son los NFTS y cómo funcionan? - <https://www.xataka.com/basics/que-nft-como-funcionan> - 09/10/2023
- [7] Metadatos - <https://www.docunecta.com/blog/que-son-los-metadatos> - 09/10/2023
- [8] Tokenizado - <https://www.plainconcepts.com/es/tokenizacion/> - 09/10/2023
- [9] Hablemos de blockchain: ¿Que son los NFT? - <https://www.sabermassermas.com/hablemos-de-blockchain-que-son-los-nft/> - 09/10/2023
- [10] Cripto-Monedero - <https://www.businessinsider.es/cripto/monedero-criptomonedas/> - 09/10/2023
- [11] Guía nfts – Que son los NFT - <https://www.socios.com/es-es/guia-nfts/que-son-los-nft/> - 09/10/2023
- [12] Skins - <https://www.geekno.com/glosario/skin> - 10/10/2023
- [13] Ventajas y desventajas nfts - <https://www.coleccionnft.es/ventajas-desventajas-nfts/> - 10/10/2023
- [14] NFT User Number - <https://www.statista.com/statistics/1266322/nft-user-number/> - 11/10/2023
- [15] OpenSea - <https://opensea.io/es> - 13/10/2023
- [16] Atomic Market - <https://atomicmarket.io/> - 13/10/2023

- [17] Rarible - <https://rarible.com/> - 13/10/2023
- [18] SuperRare - <https://superrare.com/> - 13/10/2023
- [19] Marketplaces to purchase nft - <https://geekflare.com/es/marketplaces-to-purchase-nft/> - 13/10/2023
- [20] Binance Smart Chain - <https://www.bnbchain.org/en/bnb-smart-chain> - 13/10/2023
- [21] Polygon - <https://polygon.technology/> - 13/10/2023
- [22] Solana - <https://solana.com/es> - 13/10/2023
- [23] JSON - <https://blog.hubspot.es/website/que-es-json#:~:text=JSON%20es%20un%20formato%20de,una%20alternativa%20al%20lenguaje%20XML.> - 13/10/2023
- [24] IOS - <https://www.gabit.org/gabit/?q=es/que-es-ios> - 13/10/2023
- [25] SWIFT - <https://www.apple.com/la/swift/> - 13/10/2023
- [26] Android Studio - <https://developer.android.com/studio/intro?hl=es-419> - 14/10/2023
- [27] Desarrollo aplicaciones con Kotlin - <https://platzi.com/blog/kotlin-desarrollar-apps-android/> - 14/10/2023
- [28] ¿Que es Kotlin y para qué sirve? - <https://www.plainconcepts.com/es/kotlin-android/> - 14/10/2023
- [29] Que es Figma - <https://cei.es/que-es-figma/> - 15/10/2023
- [30] Sourcetree - <https://www.sourcetreeapp.com/> - 17/20/2023
- [31] Sourcetree Tutorial - <https://soporte.code-fu.net.ni/sourcetree-tutorial/#:~:text=%C2%BFQu%C3%A9%20es%20SourceTree%3F,usuario%20que%20provee%20la%20aplicaci%C3%B3n.> - 17/10/2023
- [32] Arquitectura de aplicaciones móviles - <https://appmaster.io/es/blog/arquitectura-de-aplicaciones-moviles-en-2022-construya-su-aplicacion-movil> - 17/10/2023
- [33] Introduction a las Clean Architectures - <https://medium.com/@diego.coder/introducci%C3%B3n-a-las-clean-architectures-723fe9fe17fa> - 17/10/2023
- [34] Clean Architecture Beginners guide - <https://betterprogramming.pub/the-clean-architecture-beginners-guide-e4b7058c1165> - 17/10/2023

- [35] Clean Architecture - <https://www2.deloitte.com/es/es/pages/technology/articles/clean-architecture.html> - 17/10/2023
- [36] MVVM - <https://keepcoding.io/blog/que-es-el-patron-de-arquitectura-mvvm/> - 20/10/2023
- [37] Que es el patrón de arquitectura MVVM - <https://keepcoding.io/blog/que-es-el-patron-de-arquitectura-mvvm/> - 20/10/2023
- [38] Fragment - <https://developer.android.com/guide/fragments?hl=es-419> - 02/11/2023
- [39] Android MVVM Live data Data Binding - <https://www.digitalocean.com/community/tutorials/android-mvvm-livedata-data-binding> - 02/11/2023
- [40] Basic Kotlin Livedata Training - <https://developer.android.com/codelabs/basic-android-kotlin-training-livedata?hl=es-419#0> - 02/11/2023
- [41] Dependencies - <https://developer.android.com/studio/build/dependencies?hl=es-419>
- [42] Gradle - <https://gradle.org/> - 03/11/2023
- [43] Cloudinary - https://cloudinary.com/blog/addressing_mobile_challenges_with_the_new_cloudinary_sdk_for_android - 03/11/2023
- [44] Android X - <https://developer.android.com/jetpack/androidx?hl=es-419> - 03/11/2023
- [45] How Amplify Works - <https://docs.amplify.aws/android/how-amplify-works/> - 03/11/2023
- [46] HTTP - <https://developer.mozilla.org/es/docs/Web/HTTP/Overview> - 04/11/2023
- [47] Que es Api Rest - <https://blog.hubspot.es/website/que-es-api-rest> - 04/11/2023
- [48] Que es el ciclo de vida de un proyecto : <https://www.atlassian.com/es/work-management/project-management/phases#:~:text=El%20ciclo%20de%20vida%20de,%2C%20ejecuci%C3%B3n%20supervisi%C3%B3n%20y%20cierre.> - 08/11/2023
- [49] Pros y Contras de la metodología cascada - <https://www.lucidchart.com/blog/es/pros-y-contras-de-la-metodologia-de->

[cascada#:~:text=El%20m%C3%A9todo%20de%20cascada%20se,para%20cambios%20o%20revisiones%20imprevistos.](#) - 03/12/2023

[50] Modelo en cascada - <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/> - 04/12/2023

[51] ¿Qué es un diagrama de actividades? - <https://www.servicenow.com/es/workflows/creator-workflows/what-is-activity-diagram.html> - 04/12/2023

[52] Casos de uso - <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/casos-de-uso/> - 04/12/2023

