UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA

TESIS DE GRADO

"CRYPTO JUEGO BASADO EN BLOCKCHAIN PARA COLECCIONAR AVES VIRTUALES DE LA FAUNA BOLIVIANA"

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA

MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE: PAUCARA TORREZ RICARDO

TUTOR: M. SC. ROSA FLORES MORALES

LA PAZ – BOLIVIA

2022

HOJA DE CALIFICACIONES

UNIVERSIDAD MAYOR DE SAN ANDRÉS

FACULTAD DE CIENCIAS PURAS Y NATURALES

CARRERA DE INFORMÁTICA

Tesis de grado

SISTEMA AUTOMATIZADO DE EVALUACIÓN DE FORMATO EN TESIS Y PROYECTOS DE GRADO

Presentado por: Ricardo PaucaraTorrez

Para optar el grado Académico de Licenciado en Informática

Mención Ingeniería de Sistemas Informáticos

Nota Numeral: ……

Nota Literal: ……

Ha sido: ……...

Director de la carrera de Informática: Lic. Rubén Alcón

Tutora:	M.Sc. Rosa Flores Morales
Presidente tribunal:	Ph.D. Tapia Baltazar Jose Maria
Tribunal:	M.Sc. Clavijo Cardenas Edgar Palmiro
Tribunal:	M.Sc. Huanca Ticona German
Tribunal:	M.Sc. Vargas Blacutt Roberto

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y
NATURALES CARRERA DE INFORMÁTICA

LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

a) Visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
b) Copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
c) Copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo lareferencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

Dedicatoria

Dedico este trabajo a mi padre Pedro, a mi madre Lucia y mi hermano Efrain.

Agradecimientos

Quiero agradecer a mi familia por todo. Este proyecto pude hacerlo gracias a ustedes, gracias de verdad por todo su trabajo, su esfuerzo y su vida.

Agradezco a mi tutora M. Sc. Rosa Flores Morales por su paciencia conmigo, y por inculcarme el espíritu científico.

Agradezco a todas las personas que me dieron su amistad, me enseñaron mucho, gracias amigos.

ricardo.paucara.torrez@gmail.com

Resumen

En esta tesis se propone la implementación de un crypto juego de colección de tarjetas vinculadas a un token en una cadena de bloques, cuyo objetivo tiene el de poder ayudar a difundir el conocimiento y valoración de la diversidad de aves en territorio boliviano, se construye sobre la red blockchain ethereum, la cual por medio de sus características, se implementará el estándar de contrato inteligente ERC721 tecnología de los token no fungibles (NFT), que se utiliza para darle propiedad digital a los usuarios sobre las aves del juego. Para el desarrollo del prototipo se hizo uso de la metodología adaptada para el desarrollo de videojuegos SUM.

Palabras clave: Cadena de bloques, Ethereum, Contrato Inteligente, Crypto Juego, Propiedad Digital, ERC721, Token No Fungible.

Abstract

This thesis proposes the implementation of a crypto card collection game linked to a token on a blockchain, the objective of which is to help spread the knowledge and appreciation of the diversity of birds in Bolivian territory, it is built on the ethereum blockchain network, which through its characteristics will implement the ERC721 smart contract standard, non-fungible token technology (NFT), which is used to give users digital ownership of game birds. For the development of the prototype, the methodology adapted for the development of SUM video games was used.

Keywords: Blockchain, Ethereum, Smart Contract, Crypto Game, Digital Property, ERC721, Non-Fungible Token.

Índice

Capítulo I

Marco Introductorio

1.1 Introducción

> Cada una de las aves presentes en el país tiene una función trascendental en todas las ecorregiones y se las puede considerar un grupo taxonómico idóneo para la identificación de problemas de conservación y para la determinación de la salud de los ecosistemas (Maillard, 2012, p.2).

Con el transcurso del tiempo surgen innovaciones tecnológicas de gran impacto como lo es Blockchain, estos nuevos recursos tecnológicos son usados para todo ámbito, sin duda alguna podemos aplicarlo para apoyar a frenar la perdida de biodiversidad de nuestro planeta.

La integración de la tecnología Blockchain en videojuegos permite una propiedad comunitaria de los juegos, da una posibilidad única de monetización para crear experiencias más ricas y profundas.

Actualmente, la industria del videojuego y la tecnología blockchain aún está en un campo investigativo, los cripto juegos están en un periodo de auge y relativamente es un área poco explorada en nuestro medio. Según Scholten (2019), un cripto-juego se entendería por juegos que han conectado artículos del juego (activos) a un token, y estos activos pertenecen a una red blockchain, por ejemplo ethereum. Esto permite a través de un contrato inteligente el intercambio seguro de los artículos entre sus jugadores, y a su vez poder intercambiar por una moneda regular. Los juegos de colección de tarjetas se encuentran entre los más populares en el espacio de los token no fungibles (NFT), estos

activos en los cripto juegos pueden subir su valor dependiendo la rareza de la tarjeta, o la importancia en la narrativa.

Este trabajo presenta un videojuego el cual se desarrolla en torno a la temática de la diversidad de aves, considerando que Bolivia es el quinto país en el mundo con mayor diversidad de especies. El fin ulterior de este trabajo es el de difundir el conocimiento y promover la valoración por las aves de nuestro país, para ello se desarrolla un cripto juego de colección de tarjetas de aves vinculadas a los NFTs utilizado la tecnología blockchain.

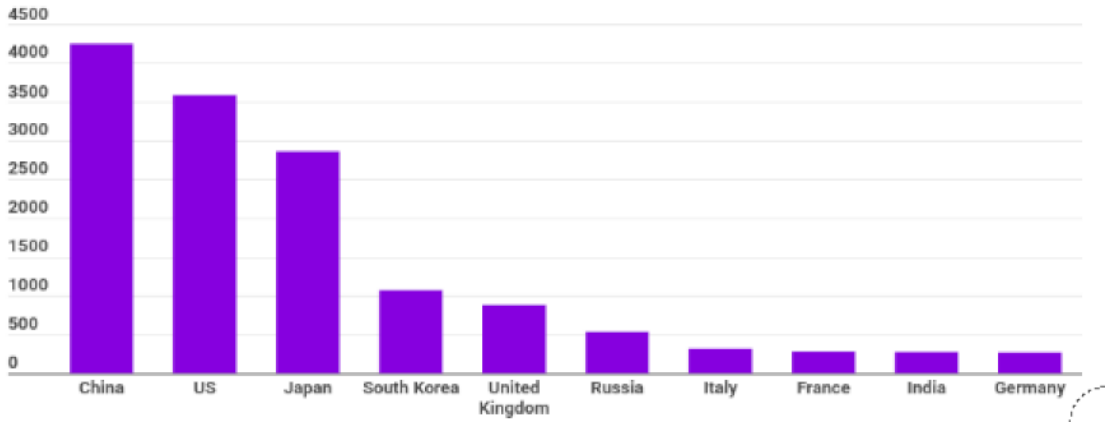
1.2. Antecedentes

Videojuegos y Blockchain

Según Putra (2020), la industria de los videojuegos siempre es y ha sido impulsora de nuevas tecnologías, como Inteligencia Artificial (IA), Realidad Virtual (VR), Realidad Aumentada (AR). Esto atrae a todo tipo de público, siendo una industria de miles de millones y va creciendo cada año (ver figura 1.1), convirtiéndose en la industria cultural líder a nivel global, creando en este ecosistema todo tipo de modelos económicos.

Figura 1.1

Ingresos globales de juegos en línea 2019(en millones de dólares)



Nota. Imagen tomada de Ingresos de juegos, 2020, (goldencasinonews.com)

CryptoKitties.

Es un juego de blockchain en Ethereum desarrollado por Axiom Zen, los usuarios recolectan y crían criaturas que llamamos cryptoKitties, cada gatito tiene un genoma único que define su apariencia y rasgos, los jugadores pueden criar a sus gatitos para crear nuevos gatos y desbloquear atributos raros.
Es uno de los primeros juegos de blockchain del mundo. Si bien cryptoKitties no es una moneda digital, ofrece la misma seguridad: cada CryptoKitty es único y 100% de su propiedad. No se puede replicar, quitar ni destruir.(cryptokitties.co,2017).

CryptoPunks.

Se lanzó en junio de 2017 como uno de los primeros tokens no fungibles (NFT) en la cadena de bloques Ethereum.
Los CryptoPunks son 10,000 caracteres generados de forma única. No hay dos exactamente iguales, y cada uno de ellos puede ser propiedad oficial de una sola persona en la cadena de bloques Ethereum. Originalmente, cualquiera que tuviera una billetera Ethereum podría reclamarlos de forma gratuita, pero los 10,000 se reclamaron rápidamente. Ahora deben comprarse a alguien a través del mercado que también está integrado en la cadena de bloques. A través de este mercado puedes comprar, pujar y ofrecer punks a la venta. (larvalabs.com, 2020).

Trabajos similares en la carrera de Informática

En el repositorio de la carrera e informática, se encuentran los siguientes trabajos relacionados con la temática del presente proyecto.

- ❑ “Certificación digital de títulos académicos basado en contratos inteligentes de la tecnología Blockchain”, desarrollado por Jaime Jesus Alvarado Perez, en el año

2020, propone desarrollar una aplicación descentralizada de emisión y autenticación de certificados de títulos académicos digitales para brindar mayor confiabilidad.

- ❑ "Desarrollo del sistema de gestión, administración y validación de la plataforma de registro de aves en áreas urbanas. Caso: Ciudad de La Paz", desarrollado por Nelson Fidel Quispe en el año 2015, propone desarrollar un sistema automatizado y continuo de gestión, administración y validación para la plataforma de registro de aves.
- ❑ "Sistema de georreferenciación on Line de registro de aves urbanas. Caso: Ciudad de La Paz", desarrollado por Alex Enrique Santos Quispe en el año 2015, propone desarrollar e implementar un Sistema de Información Georreferencial On Line que permita el registro de datos y brinde información confiable de las diferentes especies de aves observadas en nuestra ciudad y sus alrededores.

1.3. Planteamiento de la problemática

Las aves han sido motivo de fascinación, y por supuesto no sólo para admirarlas, sino hasta para envidiarlas. Las aves son apreciadas tanto por su valor estético como también por aportar el bienestar humano con la provisión de servicios ecosistémicos extraordinarios, según Maillard (2020), el grupo de frugívoros como las parabas y loros cumplen un importante papel en la dispersión de semillas de especies arbóreas y palmeras. Las especies carroñeras, como los suchas y cóndores tienen la función sanitaria de limpiar desechos orgánicos, ayudando al cuidado de la salud ambiental. Los picaflores se encargan de la polinización de muchas de las especies de plantas. La unión de esfuerzos es vital para la conservación de nuestras aves y toda forma de vida en Bolivia y el mundo.

Según Maillard (2012), se estima que las actividades del ser humano están provocando que se pierdan especies en un ritmo entre cien a mil veces más rápido de lo que se consideraría como normal.

1.3.1. Problema Principal

No se aprovecha la popularidad de los criptojuegos para difundir la información de la diversidad de aves que habitan en nuestro territorio

1.3.2. Problemas Secundarios

- Desconocimiento de la gran variedad de especies de aves en nuestro país.
- La importancia sobre el valor artístico y estético de la imagen visual de las aves en el videojuego.

1.4. Objetivo

Desarrollar un cryptojuego de colección de tarjetas que ayude a difundir información de la diversidad de aves en Bolivia, utilizando la tecnología blockchain ethereum.

1.4.2. Objetivos Específicos

- Diseñar aves digitales de la fauna Boliviana.
- Aplicar contratos inteligentes ERC721 para unir las aves digitales a blockchain.
- Desarrollar una aplicación web para la parte visual del videojuego.
- Vincular la aplicación web con la red blockchain utilizando la billetera Metamask.

1.5. Justificación

La aplicación de nuevas narrativas en el desarrollo de videojuegos, puede ayudar a difundir y promover la valoración de nuestras aves en Bolivia.

1.5.1 Justificación Económica

Con Blockchain se puede impulsar un cambio de paradigma en la construcción de economías de juego, conceder la propiedad inmutable de los elementos del juego, resolver el robo de elementos debido a la piratería y la venta de activos, asociar los activos a los jugadores en lugar de los juegos, protegiendo así a las inversiones de tiempo, dinero que los jugadores han hecho, independiente de las decisiones del desarrollador.

1.5.3. Justificación Social

A través de este proyecto se busca aportar al fortalecimiento de iniciativas tecnológicas en las áreas de desarrollo de videojuegos, incentivar a la adopción de la tecnología blockchain, y dar un aporte a la literatura sobre desarrollo de videojuegos con nuevas tecnologías en nuestro medio.

1.6. Metodología

Para lograr los objetivos del estudio y sus posibilidades se utilizó la metodología SUM, específica para gestionar el desarrollo de videojuegos que basa sus principios en metodologías ágiles, utilizando Scrum y Extremme Programming como base en su construcción (Torrez, 2016).

1.7. Alcance

El videojuego integrado a blockchain contempla los siguientes alcances

- Sorteo de aves random cada 24 horas gratis.
- Las aves están vectorizadas y vinculadas a un NFT.
- Cada usuario tendrá un álbum web para ver su colección de aves.
- Los usuarios tienen 100% de propiedad sobre sus tarje

Capítulo II

Sobre las Aves

> Alrededor del mundo y prácticamente en todos los hábitats, incluso los que ha generado (o degenerado) el hombre, vuelan, corren y nadan, aproximadamente 10.000 especies de bípedos con plumas, cumpliendo con trascendentales roles que permiten que la vida siga prosperando en este planeta azul (Rojas et al, 2015).

Las aves en el planeta tienen relevancia por diversos aspectos, según Balderrama (2009), desde el punto de vista ecológico pues actúan como dispersoras de semillas, polinizadoras, reguladoras de algunas poblaciones de insectos, gasterópodos y artrópodos. Si las aves desaparecieran de la Tierra, el equilibrio de las áreas naturales se alteraría y muchas especies de animales y plantas tenderían a desaparecer.

Según Rojas et al. (2015), en Bolivia con aproximadamente 1422 especies de aves, se encuentra a nivel mundial entre los cinco países con mayor diversidad en aves, cuenta con 15 especies endémicas.

2.1 Aves amenazadas de Bolivia

Existen una serie de factores implicados en la disminución mundial de las poblaciones de aves, dos se consideran de importancia primordial: la pérdida de hábitat y la captura de individuos para el comercio de mascotas. Según el Ministerio de Medio Ambiente y Agua de Bolivia (2009), de las 1.422 especies de aves presentes en el país, 51 se encuentran bajo alguna categoría de amenaza (Rojas et al, 2015).

2.2 Categorías de conservación

Según Rojas et al. (2015), cada especie representa un caso particular en cuanto a las presiones sobre ella y los recursos de los cuales depende. En tal sentido, con el fin de

priorizar esfuerzos de conservación es necesario clasificarlas según el grado y el tipo de amenaza que sufre cada una. Estas categorías, junto con información acerca del tamaño poblacional, distribución y tendencia de la especie determinan su *estatus* (estado) de conservación.

Entonces para poder saber cómo anda una especie en cuanto a su *estatus* de conservación es necesario investigar su distribución a diferentes escalas (dónde existe y dónde ya no), su tamaño poblacional y qué amenazas pueden tener y cuán graves son. La organización mundial que gestiona toda esta información y oficialmente ubica a especies en categorías de conservación es la Unión Internacional de Conservación para la Naturaleza (UICN). Las categorías más importantes son las siguientes:

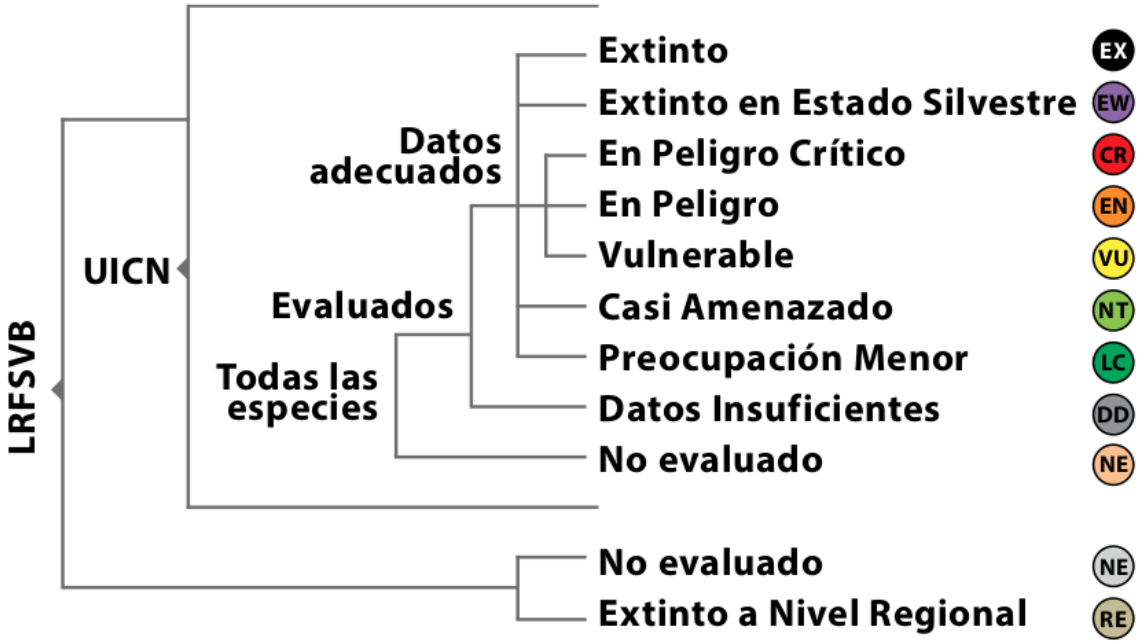
- **Peligro Crítico (CR)**. Son especies que están al borde de la extinción. Por lo general sólo existen en un único lugar del planeta, tienen una muy baja población que continúa en declive y sufren de una o más amenazas fuertes. Las especies muy raras caen en esta categoría.
- **En Peligro (EN)**. Son especies que aún no están en la categoría crítica, pero que pueden pasar a estarlo pronto si no se toman acciones urgentes de conservación. Nuevamente, esta categoría reúne a las especies raras, especialmente loros y parabas, que además de pérdida de hábitat sufren también de captura para el comercio de mascotas.
- **Vulnerable (VU).** Son especies que aún se mantienen, pero que son muy frágiles por tener una población baja. Pueden estar amenazadas por hábitat intervenido o algunas otras fuentes de presión.

Las restantes categorías de conservación son: Dependiente de Conservación, Casi Amenazado, Menor Riesgo y Deficiente en Datos. Esta última categoría reúne a todas las

especies cuyas poblaciones no han sido bastante estudiadas como para establecer su estado de conservación. (Figura 2.1)

Figura 2.1

Categorías de conservación de la UICN



Nota. Diagrama tomado de *Aves Amenazadas del departamento de Santa Cruz*, 2015 (p.10) por Rojas et al. (2015)

Según afirma Balderrama (2015), se evaluaron 148 especies de aves en Bolivia, de ellas tres quedaron categorizadas como Críticamente Amenazadas (CR), 10 en Peligro (EN), 40 vulnerables (VU), 89 casi amenazadas (NT) y seis especies como datos deficientes (DD), un total de 53 especies de aves que presentan algún grado de amenaza (Tabla 2.1).

Tabla 2.1

Tabla de aves con algún grado de amenaza de Bolivia

Familia	Especie	Categoría 1996 (Rocha & Quiroga, 1996)	Categoría 2003 (Rocha et al. 2003)	Categoría 2008 (Balderrama, 2008)
EN PELIGRO CRÍTICO				
Cracidae	Crax globulosa	VU	VU	CR
Cracidae	Pauxxi unicornis	VU	EN	CR
Psittacidae	Ara glaucogularis	CR	CR	CR
Psittacidae	Ara rubrogenys	EN	EN	CR
Furnariidae	Cinclodes aricomae	CR	CR	CR
Formicariidae	Hylopezus auricularis	NE	VU	CR
Cotingidae	Phibalura Boliviana	NE	NT	CR
EN PELIGRO				
Rheidae	Rbea pennata	EN	EN	EN
Podicipedidae	Rollandia microptera	VU	VU	EN
Accipitridae	Harpybaliaetus coronatus	VU	VU	EN
Caprimulgidae	Caprimulgus Candicans	EN	EN	EN
Furnariidae	Craniolenca Benricae	NE	EN	EN
Furnariidae	Asthenes urubambensis	NE	NT	EN
Thamnophilidae	Terenura sbarpei	VU	EN	EN
Tyrannidae	Anairetes alpinus	CR	CR	EN
Tyrannidae	Cnemarchus erytbropygius	NE	NE	EN
Emberizidae	Poospiza garleppi	EN	EN	EN
VULNERABLE				
Tinamidae	Nothoprocta Taczanowskii	NE	VU	VU

Cracidae	Penelope dabbenei	VU	NE	VU
Cracidae	Chamaeopetes goudotti	DD	NE	VU
Phoenicopteridae	Phoenicoparrus andinus	VU	VU	VU
Phoenicopteridae	Phoenicoparrus jamesi	VU	VU	VU
Cathartidae	Vultur grypbus	VU	NT	VU
Accipitridae	Harpia barpyja	VU	NT	VU
Accipitridae	Oroaetus isidori	VU	NE	VU
Rallidae	Fulica gigantea	NE	NE	VU
Rallidae	Fulica cornuta	EN	EN	VU
Psittacidae	Anodorbynchus byacinthinus	EN	EN	VU
Psittacidae	Primolius couloni	NE	NE	VU
Psittacidae	Amazona tucumana	VU	DD	VU
Psittacidae	Myiopsitta lucbsi	NE	NE	VU
Strigidae	Megascops marsballi	NE	NE	VU
Furnariidae	Lepthastenura yanacensis	VU	NT	VU
Furnariidae	Astbenes berlepscbi	VU	NT	VU
Formicariidae	Grllaria andicolus	NE	NE	VU
Tyrannidae	Phyllomyias weedeni	NE	NE	VU
Cotingidae	Lipaugus arupygialis	NE	VU	VU
Cincliade	Cinclus scbulzi	VU	VU	VU
Thraupide	Oreomanes fraseri	VU	NT	VU
Emberizidae	Prophyrospiza caerulescens	NE	NT	Vu
Emberizidae	Poospiza baeri	NE	Vu	VU

Nota, està lista fue publicada en el 2008 por el Ministerio de medio ambiente y agua de Bolivia, en el libro rojo de la fauna silvestre de vertebrados de Bolivia.

2.3 Las aves en la cultura

Las aves siempre han tenido un lugar especial en las tradiciones del ser humano; han sido la inspiración para identificar deidades como como las egipcias Horus, Montu, Rha que son representadas con la cabeza de un halcón, y Tot representada con la cabeza de un Ibis o Nejbet su diosa buitre.

Las aves también han estado y están presentes en todas las manifestaciones humanas; presentes en canciones, poemas, cuentos y obras de arte. En cualquier parte del mundo la gente tiene conocimientos y aprecio especial por las aves, en muchos sitios existen pueblos cuya identidad cultural se relaciona con estas, lo que se ve reflejado en los símbolos nacionales de muchos países; para el caso de Bolivia el majestuoso Cóndor de los Andes está representado en el escudo patrio. (Rojas et al. 2015).

Capítulo III

Marco Teórico

Este capítulo describe la literatura utilizada como base para la investigación del presente proyecto. Se describen los conceptos clave sobre la tecnología Blockchain y la terminología utilizada, como también la metodología SUM para aplicarla en el siguiente capítulo.

3.1 P2P *(Peer-to-peer)*

Una red P2P, o peer-to-peer, es una red donde un grupo de personas o máquinas participan de forma completamente descentralizada. Es decir, es una red donde no hay un punto central de conexión o control, y donde las partes actúan de forma autónoma respondiendo a un protocolo de comunicaciones y consenso común. De esta forma, los integrantes de la red pueden intercambiar información de forma directa y sin intermediarios. (academy bit2me, 2017).

Para lograr este funcionamiento, las redes P2P se construyen sobre protocolos que se ejecutan sobre los protocolos de Internet (también conocido como TCP/IP). De allí que, a los protocolos P2P se le denomine protocolos de aplicación o Layer 7, según el modelo Open Systems Interconnection u OSI. Esto significa que los protocolos P2P necesitan para su funcionamiento, el uso de otros protocolos más abstractos a los fines de poder funcionar, pero que, al mismo tiempo, los hace más sencillos de construir y hacer funcionar.(academy by bit2me, 2017).

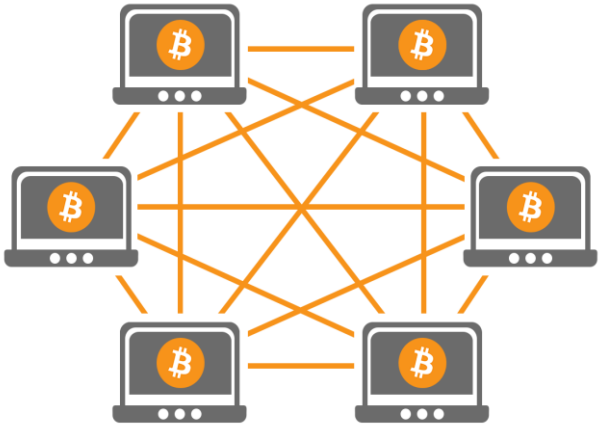
3.1.1 Tipos de redes P2P

Entre los tipos de redes P2P existentes podemos especificar los siguientes:

- Red descentralizada y estructurada. Este tipo de redes son conocidas como redes P2P híbridas. En este tipo de redes no existe un directorio en un servidor central, sino que en su lugar existen una serie de nodos o peers, que tienen la capacidad de recibir peticiones de información y responder a las mismas para facilitar el acceso a los recursos. Para evitar la centralización de esta funcionalidad, los nodos o peer especiales pueden ser instalados y configurados por cualquier persona, buscando con ello que la misma comunidad de usuarios extienda la funcionalidad de la red y permita su correcto funcionamiento.
- Red descentralizada y no estructurada. En este tipo de redes P2P no existen ordenadores o nodos que funcionan como controladores centrales de peticiones. Por el contrario, cada nodo dentro de la red tiene las mismas funciones que el resto de nodos, por lo que cada nuevo nodo ejerce la misma autoridad que el resto (Figura 3.1). En este punto, redes como Bitcoin cumplen con esta características, puesto que cada nodo conectado tiene las mismas capacidades que el resto.

Figura 3.1

Diagrama de una red Peer-to-peer



Nota. El diagrama representa una red peer to peer totalmente descentralizada. Tomada de Criptomo, 2017 (https://criptomo.com/)

3.2 Criptografía

La criptografía es la práctica de desarrollar protocolos que impiden que terceros vean datos privados. La criptografía moderna combina las disciplinas de matemáticas, informática, física, ingeniería y más. Algunos términos importantes se definen a continuación:

- **Cifrado**: codificación de texto en un formato ilegible.
- **Descifrado**: reserva de cifrado: conversión de un mensaje desordenado a su forma original.
- **Cipher**: un algoritmo para realizar el cifrado o descifrado, generalmente un conjunto bien definido de pasos que se pueden seguir.

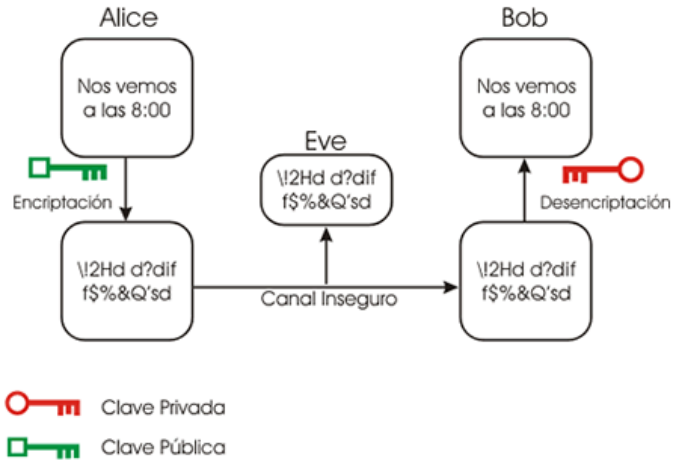
La criptografía antes de la era moderna era sinónimo de encriptación: el proceso de convertir información de un formato legible a algo que no tiene sentido. Las técnicas de cifrado se remontan a los antiguos egipcios y tienen raíces a lo largo de la historia. (Lai, 2018).

3.2.1 Criptografía de clave pública

La criptografía de clave pública (también llamada criptografía asimétrica) es un sistema criptográfico que utiliza un par de claves: una clave pública y una clave privada. La clave pública puede estar ampliamente distribuida, pero la clave privada debe ser conocida solo por su propietario. Las claves siempre se crean en un par: cada clave pública debe tener una clave privada correspondiente (Lai, 2018).(Figura 3.2).

Figura 3.2

Esquema de criptográfica de clave pública



Nota, cuando Alice hace la encriptaciòn usa la clave pùblica de Bob. Tomada de Sistema de clave pùblica, 2020 (textoscientificos.com//criptografia/clave-privada.gif)

3.2.2 Hash criptográfico

Hashing es un término informático que significa tomar una cadena de entrada de cualquier longitud y producir una salida de longitud fija. No importa si la entrada a una determinada función hash es de 3 o 100 caracteres, la salida siempre tendrá la misma longitud (Lai, 2018).

Las funciones hash criptográficas tienen las siguientes propiedades:

- **Determinista:** no importa cuántas veces le dé a la función una entrada específica, siempre tendrá la misma salida.
- **Irreversible:** es imposible determinar una entrada a partir de la salida de la función.
- **Resistencia a colisiones**: dos entradas nunca pueden tener la misma salida.

Otra característica importante de las funciones hash criptográficas es que cambiar cualquier bit de datos en la entrada altera drásticamente la salida. Por ejemplo, las salidas hash de 111111 y 111112 serían completamente únicas y no tendrían relación entre sí.

3.3 Blockchain

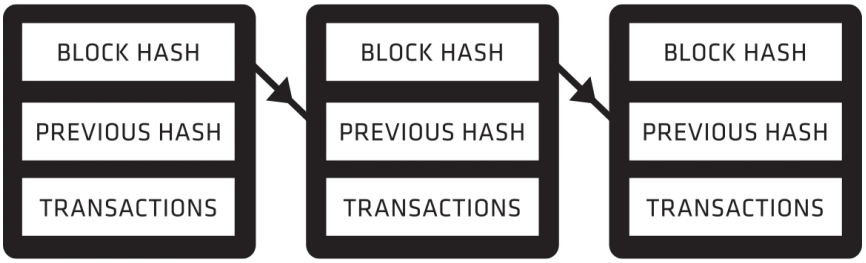
Blockchain es una base de datos distribuida, se describe mejor como una base de datos pública que se actualiza y se comparte entre muchas computadoras en una red (ethereum docs, 2021).

También se puede decir que blockchain es un método de almacenamiento de datos descentralizado, consiste en nodos que están conectados entre sí en una determinada red, en él hay registros de transacciones de datos en forma de bloque formando una cadena de bloques (Figura 3.3). La cadena de bloques actúa como una base de datos distribuida o un libro mayor global que mantiene registros de todas las transacciones en la red Blockchain. Las transacciones se sellan según el tiempo de procesamiento y se convierten en bloque, donde cada bloque se identifica por su hash criptográfico (Putra, 2020). Esta red de computadoras administra la cadena de bloques juntos sin jerarquía, su arquitectura se denomina red *peer-to-peer*.

Entonces, en esencia, la tecnología blockchain es una herramienta de mantenimiento de registros. Las palabras clave más importantes de entender son: sin terceros, sin intermediarios y descentralizado. Es una infraestructura digital.

Figura 3.3

Esquema de cadena bloques



3.3.1 Principios de la Blockchain

La infraestructura digital blockchain está sostenida por tres pilares que resumen todo su potencial.

Descentralización

En lugar de tener una autoridad de confianza única que centralice todos los registros y esté a cargo de todas las validaciones, la blockchain distribuye la información a través de la red. Para forjar la cadena de bloques, más del 51% de la potencia de procesamiento de la red es necesaria, lo que significa que el costo de lograr tal potencia informática supera en gran medida los beneficios obtenidos de ella. Esto evita que la red sea controlada por una entidad única.

Inmutabilidad

Cada registro en una cadena de bloques es permanente, cualquier intento de cambiar el historial, será rechazado y cualquier nodo que se desvíe del consenso será excluido de la red.

Transparencia

Todas las transacciones se registran y almacenan en cada nodo y cualquier participante puede ver y comprobar cualquier transacción en cualquier momento.

3.4 Ethereum

Ethereum es una plataforma de software libre construida sobre tecnología blockchain por la comunidad informática que permite a los desarrolladores construir e implementar aplicaciones descentralizadas.

Ethereum tiene como objetivo cambiar el funcionamiento de Internet, porque por primera vez, permite que los sistemas informáticos en línea se ejecuten sin utilizar ningún tercero. Ethereum permite que las aplicaciones de software se ejecuten en una red de muchas computadoras privadas (también conocido como sistema distribuido) (District0x, 2020).

3.4.1 Máquina virtual Ethereum (EVM)

La máquina virtual Ethereum es la computadora virtual global cuyo estado cada participante en la red Ethereum almacena y acepta. Cualquier participante puede solicitar la ejecución de código arbitrario en el EVM; La ejecución de código cambia el estado del EVM (ethereum docs, 2021).

3.4.2 Contrato Inteligente

La función básica que impulsa las aplicaciones y programas creados en Ethereum se llama contratos inteligentes. Los contratos inteligentes son simplemente programas, una colección de código(sus funciones) y datos (su estado) que reside en una dirección especifica en la cadena de bloques de Ethereum. (district0x, 2020).
Los contratos inteligentes de Etherum se pueden escribir con *solidity* o *viper*

3.4.3 Ether

Es una criptomoneda con abreviatura ETH, con el propósito de alimentar la red Ethereum, permitir la existencia de un mercado para la computación. Tal mercado proporciona un incentivo económico para que los participantes verifiquen y ejecuten solicitudes de transacciones y proporcionen recursos computacionales a la red (Ethereum docs, 2021).

3.4.4 Transacción

Una transacción de Ethereum se refiere a una acción iniciada por una cuenta de propiedad externa, en otras palabras, una cuenta administrada por un humano, no un contrato. Por ejemplo, si Bob envía a Alice 1 ETH, se debe debitar la cuenta de Bob y acreditar la de Alice. Esta acción de cambio de estado tiene lugar dentro de una transacción (Ethereum docs, 2021) (Figura 3.4).

Las transacciones, que cambian el estado del EVM, deben transmitirse a toda la red. Cualquier nodo puede transmitir una solicitud para que se ejecute una transacción en el EVM; después de que esto suceda, un minero ejecutará la transacción y propagará el cambio de estado resultante al resto de la red.

Las transacciones requieren una tarifa y se deben minar para que sean válidas. Los usuarios pagan la tarifa por ese cálculo, ese pago va al minero y se llama gas.

Figura 3.4

Diagrama de una transacción en el EVM

Nota, documentación ethereum 2021 (ethereum.org/es/developers/docs/transactions/)

3.4.5 Bloques

Los bloques son lotes de transacciones con un hash del bloque anterior en la cadena. Esto une los bloques (en una cadena) porque los hash se derivan criptográficamente de los datos del bloque. Esto evita el fraude, porque un cambio en cualquier bloque en el historial invalidaría todos los bloques siguientes, ya que todos los hash posteriores cambiarían y todos los que ejecutan la cadena de bloques se darían cuenta. (ethereum docs, 2021)

3.4.6 Gas y tarifas

Gas se refiere a la unidad que mide la cantidad de esfuerzo computacional requerido para ejecutar operaciones específicas en la red Ethereum.

Dado que cada transacción de Ethereum requiere recursos computacionales para ejecutarse, cada transacción requiere una tarifa. Gas se refiere a la tarifa requerida para realizar con éxito una transacción en Ethereum. (ethereum docs, 2021)

3.4.7 Minería en Ethereum

La minería es el proceso de crear un bloque de transacciones que se agregarán a la cadena de bloques Ethereum. Los mineros esencialmente procesan transacciones pendientes y reciben recompensas de bloque en forma de Ether, por cada bloque generado. La generación de un bloque requiere un trabajo computacional intensivo (o poder de hash) debido a la dificultad establecida por el protocolo de red Ethereum.(ethhub, 2020)

Ethereum, como Bitcoin, utiliza actualmente un mecanismo de consenso de prueba de trabajo (PoW). La minería es el elemento vital de la prueba de trabajo. Los mineros de Ethereum (computadoras que ejecutan software) utilizan su tiempo y poder de cálculo para procesar transacciones y producir bloques.

3.4.7.1 Prueba de trabajo

Prueba de trabajo (PoW) es el mecanismo que permite que la red Ethereum descentralizada llegue a un consenso o acuerde cosas como los saldos de las cuentas y el orden de las transacciones. Esto evita que los usuarios "gasten el doble" de sus monedas y asegura que la cadena Ethereum sea increíblemente difícil de atacar o sobrescribir. (ethereum docs, 2021)

3.5 Videojuegos

Los juegos electrónicos a menudo se llaman juegos de computadora o videojuegos. En esta disertación nosotros usamos el término videojuego, ya que tiene un significado más generalizado. Los videojuegos también se refieren a la interacción con el usuario por medio de los respectivos sistemas de entrada y salida a través de una pantalla. Los videojuegos son importantes aplicaciones informáticas y son una gran parte del

entretenimiento digital. Desde los años cincuenta, los videojuegos han pasado por diferentes fases dependiendo de inventos en tecnologías (Munir et al. 2019).

3.5.1 Tipos de videojuegos

Los videojuegos se pueden dividir en diferentes categorías dependiendo de las tecnologías en que se van construyendo y jugando, lo que se conoce como plataforma. La plataforma es la combinación de hardware y software que permite operar el videojuego.

Con la invención de diferentes tecnologías a lo largo de la historia, surgieron varios tipos de juegos que dependían de su respectiva plataforma, en la que se crearon, la tabla 3.1 muestra un resumen de este desarrollo.

Tabla 3.1

Tipos de videojuegos

Juegos Mainframe	La mayoría de los primeros videojuegos fueron los juegos de mainframe porque se jugaban en computadoras mainframe y principalmente lo jugaban investigadores y especialistas en informática
Juegos Arcade	Estos juegos son verdaderamente juegos electrónicos en lugar de juegos de computadora. Los juegos arcade se juegan en máquinas especiales que funcionan con monedas que, a diferencia de un sistema informático, pueden que no realicen otra tarea que jugar el juego. Estas máquinas especiales que funcionan con monedas fueron diseñados para jugar únicamente a los videojuegos y principalmente tenían una máquina de estado cableada construido a partir de puertas lógicas y contadores.

Juegos de consola	A diferencia de las máquinas de juegos de arcade, se inventaron consolas especiales que se pueden conectar a un televisor común en casa para jugar. Las consolas de juegos vinieron con cartuchos reemplazables para una variedad de juegos
Juegos de PC	Con la invención de las computadoras personales, incluidas TRS-80 y Apple I en 1976, nacieron los juegos de ordenador personal.
Juegos móviles	Los juegos móviles se pueden separar fácilmente de los juegos de PC o consola en términos de aporte e interacción involucrados. Con iOS, Android y otros estándares operativos de alto nivel. sistemas para dispositivos móviles, hay nuevas opciones disponibles para la interacción con el usuario como información de posicionamiento global, dispositivo de cámara y pantalla táctil, que fue no es posible en juegos de consola o PC tradicionales.
Juegos de realidad virtual	La unidad montada proporciona pantallas estereoscópicas y seguimiento de movimiento para mostrar un entorno virtual que responde con el movimiento de la cabeza del jugador.
Juegos en línea(online)	Estos se juegan en PC o consolas, pero la plataforma aquí cambia en términos de la red. Estos se reproducen a través de Internet. La mayoría de los juegos de PC eran juegos para un solo jugador y se jugaron con o contra el software del juego. Con la comercialización de internet, nació la era de los juegos en línea. Los juegos en línea los juegan uno o más personas a través de Internet en sus respectivos dispositivos, es decir, consola de juegos, computadora PC o teléfonos móviles.

Nota, tabla tomada de *Challenges and security aspects of blockchain based online multiplayer games* (p.15), por Munir et al. 2019

3.5.2 Cantidad de jugadores

Respecto a la cantidad de jugadores que interactúan en un videojuego tenemos las modalidades de *single player* (solo) y *multiplayer* (multijugador).

En solitario, *single player* significa que un solo jugador está jugando con el software del videojuego, o en contra de una IA(Inteligencia Artificial) muy comunes en juegos de estrategia. Un videojuego en solitario puede ser en línea o fuera de línea según el proveedor y el juego.

Se denomina *multiplayer* cuando un jugador puede jugar contra otro, esto hace que el juego sea más interesante, y ayudo al crecimiento de esta industria la posibilidad del modo multijugador, actualmente son mucho más populares. Aunque los juegos en solitario se juegan en todo el mundo. (Munir et al. 2019)

Los juegos multijugador vienen en tres categorías.

a) Multijugador por turnos, esta fue la fase más inicial del juego multijugador. Por turno, cada jugador toma su turno en el juego, como un juego de ajedrez con un paso a la vez o con un juego de carreras al jugar todo el nivel en cada turno y los resultados se pueden producir en función del rendimiento de cada jugador.
b) Multijugador en pantalla dividida, estos juegos son perfectos para la categoría que cuenta para la entrada en tiempo de ejecución, como carreras o juegos de supervivencia. Cada jugador ve el personaje suyo en su parte de la pantalla y están compitiendo en el mismo nivel. El único en alcanzar la meta del principio gana. De esta forma ambos jugadores compiten al mismo tiempo con o unos contra otros. En el tipo de supervivencia, cada jugador ve un holograma de su compañero u oponente.

c) Multijugador en línea, en este tipo, los jugadores pueden jugar desde diferentes lados del planeta, y compitiendo entre sí. Gracias a la tecnología de Internet, estos juegos se juegan a través de Internet y el número de jugadores puede ser de cientos a miles dependiendo de la jugabilidad. El enfoque de la investigación está en el modo multijugador en línea. Además existen diferentes tipos de juegos multijugador en línea.

3.5.3 Seguridad en videojuegos

Los activos digitales se componen de sistemas que incluyen hardware y software, comunicación de red y datos. Cualquier intento malicioso de violar estos activos se llama ciberataque. Los ataques cibernéticos se realizan para obtener beneficios a través de medios ilícitos y son amenazas importantes para organizaciones con valioso sistema de información. Posibilidad de cualquier tipo de ataque organizado, o las infracciones en el sistema de información, ya sea por error humano o por incumplimiento accidental, se definen como amenazas de seguridad cibernética.

La seguridad cibernética en los videojuegos tiene como objetivo proporcionar protección a los activos digitales, progreso del jugador en el juego, datos de información del mismo, de estos ataques cibernéticos, además de protección contra jugadores malintencionados denominados hackers que rompen con la mecánica del juego. De acuerdo con a Chen Zhao, (2008), citado por Munir et al. (2019), "La seguridad cibernética consiste en tecnologías, procesos y controles diseñados para proteger los sistemas, las redes y los datos de los ciberataques".

3.6 Blockchain y los videojuegos

Con la llegada de Blockchain, es natural buscar aplicarlo en todo tipo de actividades, los videojuegos no son la excepción.

La unión de la tecnología blockchain y los juegos es una gran promesa para el crecimiento de ambas industrias. Según Curran (2019), las innovaciones en el campo incipiente de los juegos de blockchain han superado los límites de los activos no fungibles y están preparados para seguir proporcionando desarrollos novedosos en otras áreas como la escalabilidad.

3.6.1 Blockchain y la arquitectura del juego.

A diferencia de los juegos tradicionales, en la blockchain los jugadores del videojuego deben registrar una dirección en la cadena de bloques correspondiente antes de comenzar sus sesiones de juego. La dirección de blockchain, a la que accede mediante una billetera, sirve como destino de los activos virtuales para su correspondiente jugador.

Desde la perspectiva de la cadena de bloques, admitir juegos digitales implica requisitos de alto rendimiento que incluyen transacciones por segundo (TPS), latencia de respuesta, etc.(Miny y Yang,2019)

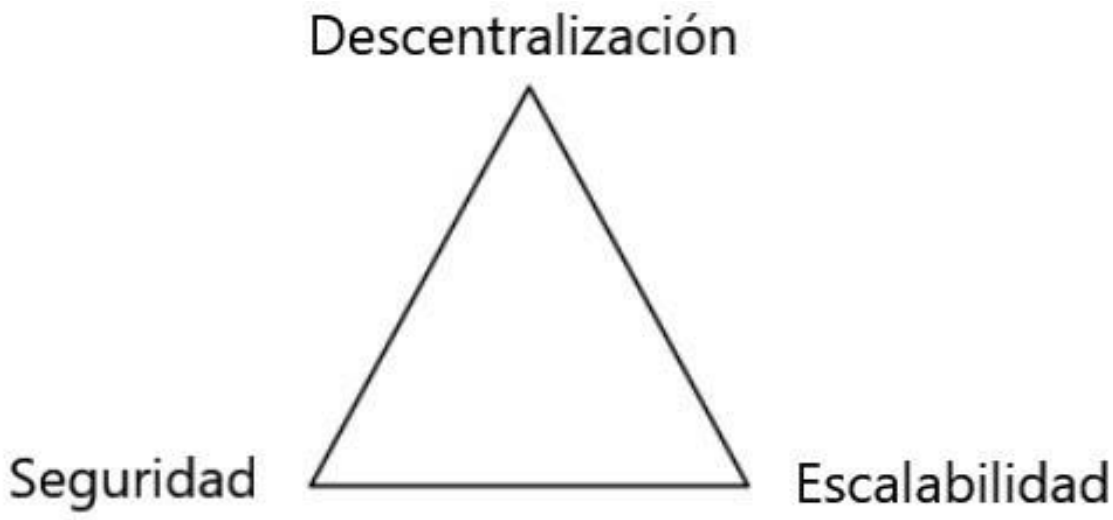
Ethereum aún no logró satisfacer estas necesidades, la arquitectura de un videojuego para la blockchain sigue siendo restringida, la jugabilidad está limitada a algunas mecánicas, casi en la totalidad de los títulos están reducidos a juegos del genero colección y recolección.

El trilema de la arquitectura en blockchain y videojuegos

En la unión de estos dos campos: Blockchain y videojuegos, está plagado de inconvenientes, siendo todavía esta unión relativamente corta, cuenta con varios problemas a la hora de ponerse a desarrollar. Esto es un problema conocido como el trilema blockchain. Al igual que un conocido problema de administración de proyectos(bueno, barato y rápido: elija 2), las dificultades de blockchain para equilibrar la descentralización, la escalabilidad y la seguridad (figura 3.5).

Figura 3.5

Trilema blockchain, equilibro: Descentralización, Escalabilidad, y Seguridad (elija 2)



Nota, elegir dos y necesaria mente descuidar el tercero.

Un problema está relacionado con los desarrolladores de juegos, según Balaguer & Caracciolo, (2019), para desafío de los desarrolladores, ¿Cómo tienen que adoptar su proceso de desarrollo? ¿Qué pasos se deben tomar?, ¿Qué problemas deben tener en cuenta antes de desarrollar tales juegos?. Seguramente habrá algunos beneficios de integrar la cadena de bloques con los videojuegos, pero hace esto ¿resolver todos los problemas de seguridad?. Porque la tecnología blockchain brinda seguridad en términos de propiedad de los activos y su comercio, pero los juegos en línea tienen el potencial de ser pirateados. Otra cuestión es el problema de la escalabilidad, ¿Es posible desarrollar juegos descentralizados que se sientan como un juego normal?, ¿Son limitadas las características que se pueden implementar aparte de la propiedad y reutilización de activos, como: mecánicas y jugabilidad?

3.6.2 Ethereum y el Cripto Juego

Un cripto juego es un videojuego tradicional con la posibilidad de crear items y asociarlos a un token en la blockchain.

Tomar un juego de red donde los jugadores pueden intercambiar oro virtual por gatos virtuales, la propiedad de cualquier elemento virtual debe registrarse en algún lugar para asegurar la persistencia del elemento dentro del juego. (Scholten et al. 2019)

Al respecto se indica:

> Tradicionalmente, el desarrollador de juegos C suele almacenar esta información en su propio servidor central de juegos. Cuando el jugador A vende un espada virtual(activo) a un jugador B en el juego, esto activa una solicitud del cliente del jugador A al servidor del proveedor de juegos C para actualizar la espada virtual entrada de propiedad en su base de datos. En el servidor del proveedor de juegos C la base de datos es un libro de contabilidad central: un único registro autorizado de transacciones. Si bien los libros centrales pueden ser eficientes, también constituyen un único punto de falla. Si una falla de red corrompe la transferencia de una transacción al servidor o un hacker manipula su base de datos, o el proveedor del juego decide uno mañana para borrar todo el traje del mundo del juego, los jugadores perderían sus trajes virtuales. Con un libro mayor distribuido, todos los nodos de la red: jugadores A, B y proveedor de juegos C: cada uno conserva una copia idéntica del libro mayor. Cada vez que alguien emite un cambio en el libro mayor, vende un traje o una espada, uno o varios nodos de la red procesan esta transacción y utilizan algún mecanismo de consenso entre todos nodos para establecer la nueva entrada correcta, que luego se copia en todos los libros de contabilidad. Por lo tanto, los participantes no tienen que confiar en ningún nodo único para no perder o corromper o manipular el libro mayor. (Scholten et al. 2019).

3.6.3 Token y Blockchain

Un token es una representación de algo en la cadena de bloques. Este algo puede ser dinero, tiempo, servicios, acciones en una empresa, una mascota virtual, cualquier cosa.

Al representar las cosas como tokens, podemos permitir que los contratos inteligentes interactúen con ellos, los intercambien, los creen o los destruyan. (OpenZeppelin, 2020)

Aunque el concepto de un token es simple, tienen una variedad de complejidades en la implementación. Debido a que todo en Ethereum es solo un contrato inteligente, y no hay reglas sobre lo que los contratos inteligentes tienen que hacer, la comunidad ha desarrollado una variedad de estándares (llamados EIP o ERCs) para documentar cómo un contrato puede inter operar con otros contratos. (OpenZeppelin, 2020)

Existen diferentes tipos de tokens.

ERC20: el estándar de token más extendido para activos fungibles, aunque algo limitado por su simplicidad.
ERC721: la solución de facto para tokens no fungibles, a menudo utilizados para coleccionables y juegos.
ERC777: un estándar más rico para tokens fungibles, que permite nuevos casos de uso y se basa en aprendizajes pasados. Compatible con versiones anteriores de ERC20.

3.6.3.1 Token No fungible

Un token no fungible, es un token criptográfico que tiene la capacidad de ser un token único e irrepetible. Uno que no puede ser dividido, pero que puede ser utilizado para representar objetos del mundo real o digital junto a sus características propias, así como la propiedad del mismo, mientras mantiene todo ello dentro de una representación en una blockchain por medio de un contrato inteligente. (academy bit2me, 2020)

Este tipo de token es perfecto para ser utilizado en plataformas que ofrecen artículos coleccionables, claves de acceso, boletos de lotería, asientos numerados para conciertos y partidos deportivos, etc. Este tipo especial de token cuenta con un estándar adecuado, el ERC- 721. (ethereum docs, 2021)

3.6.3.2 ERC-721

El ERC-721 (Ethereum *Request for Comments* 721), propuesto por William Entriken, Dieter Shirley, Jacob Evans, Nastassia Sachs en enero de 2018, es un estándar de token no fungible que implementa una API para tokens dentro de los contratos inteligentes.

Proporciona funcionalidades como transferir tokens de una cuenta a otra, obtener el saldo actual del token de una cuenta, obtener el propietario de un token específico y también el suministro total del token disponible en la red. Además de estas, también tiene algunas otras funcionalidades como aprobar que una cantidad de token de una cuenta pueda ser movida por una cuenta de terceros.(ethereum docs, 2021)

Si un contrato inteligente implementa los siguientes métodos y eventos, se puede llamar un contrato de token no fungible ERC-721 y, una vez implementado, será responsable de realizar un seguimiento de los tokens creados en Ethereum.

1. Cada token ERC-721 posee un nombre. Este campo se utiliza para indicar a los contratos y aplicaciones externas la denominación del token.
2. Tienen definido un símbolo que permite que las DApps puedan acceder a un nombre abreviado para dichos tokens.
3. Llevan definido el suministro total del token.
4. Contienen un campo que indica el balance de tokens dentro de una dirección.
5. Cada token ERC-721 lleva definido un campo de funciones del propietario, usado para definir la propiedad del token y como se puede transferir la misma.
6. Llevan definido un campo llamado Propietario, el cual permite garantizar la no fungibilidad del token e identificar criptográficamente el mismo.
7. Cuenta con un campo llamado Aprobación, mediante el cual se otorga permiso a otra entidad para transferir el token en nombre del propietario.
8. Llevan definidos también otro campo de nombre Toma de posesión, el cual permite que un usuario pueda poseer determinada cantidad de tokens y desea retirarlos del saldo de otro usuario.

9. Por otro lado, el campo de Transferencia, permite el envío de tokens a otro usuario de la misma forma que sucedería con una criptomoneda y detalla que cuenta envió el token y cual lo recibió, junto con el ID de ese token.
10. Debido a la singularidad del token y de que un usuario puede poseer diversos tokens ERC-721, se ha creado el campo Token del propietario por índice. Esta función permite hacer el seguimiento de los tokens por medio de un ID único.
11. Por último los token ERC-721 cuentan con un campo llamado Metadatos del token. Es precisamente este campo el que permite su condición de no fungibles y alberga todas esas propiedades que distinguen a un token de todos los demás.
12. Estos tokens no permiten operaciones de *allowance* en su estructura

El punto 11, el estándar erc721 requiere un campo para los metadatos a guardar en la blockchain, es donde el activo NFT guarda sus propiedades, es donde está su diferencia, donde se vuelve único e irrepetible. Según OpenZeppelin, (2021), se puede aprovechar en utilizar la tecnología de *InterPlanetary File System* (IPFS) para guardar nuestros metadatos y tener una mayor seguridad y transparencia.

3.6.3.3 InterPlanetary Files Sytem (IPFS)

Según Benet (2019), el Sistema de Archivos InterPlanetario (IPFS) es un sistema de archivos distribuido peer-to-peer que busca conectar todos los dispositivos informáticos con el mismo sistema de archivos. IPFS proporciona un modelo de almacenamiento en bloques con dirección de contenido de alto rendimiento, con hipervínculos direccionado por contenido. Esto forma un DAG Merkle generalizado, una estructura de datos sobre la que se pueden construir sistemas de archivos versionados, cadenas de bloques e incluso una Web permanente. IPFS combina una tabla hash distribuida, un intercambio de bloques incentivado y un espacio de nombres auto-certificable. IPFS no tiene un único punto de error y los nodos no necesitan confiar entre sí.

Direccionamiento de contenido

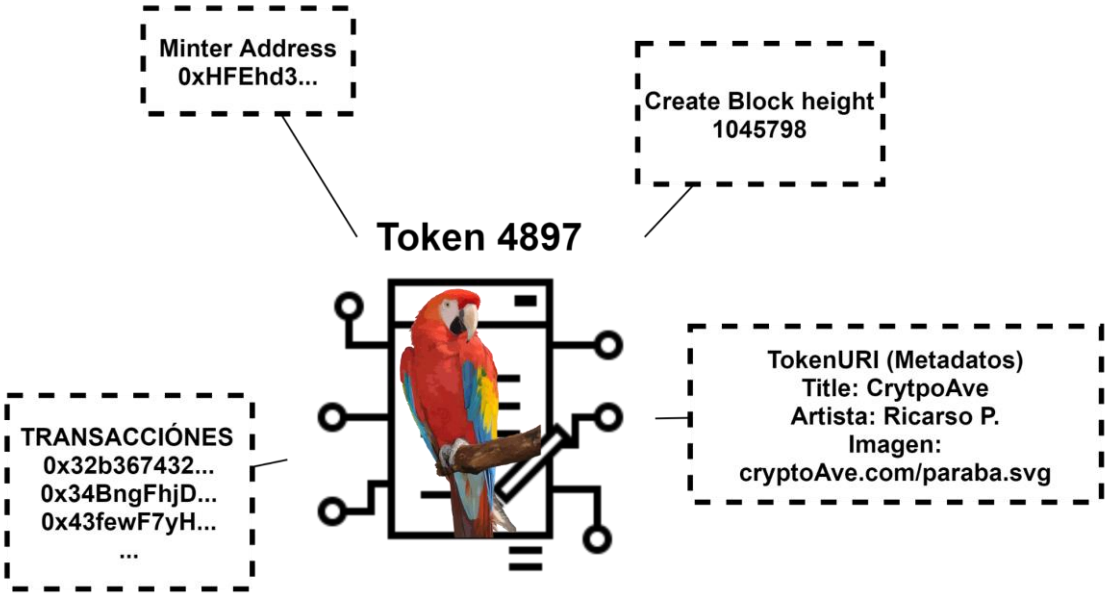
Ipfs utiliza un identificador de contenido o CID, es una etiqueta que se utiliza para apuntar al metrial en IPFS. No indica dónde se almacena el contenido, pero forma un tipo de dirección basa en el contenido en sí. Los CID se basan el hash criptográfico, como ya se describió, cualquier diferencia en el contenido producirá un CID diferente.

IPFS utiliza el algoritmo hash de forma predeterminada, pero hay compatibilidad con muchos otros algoritmos. Los CID pueden adoptar algunas formas diferentes con diferentes bases de codificación o versiones de CID. Muchas de las herramientas IPFS existentes todavía generan CID v0, aunque el (Sistema de archivos mutable) y las operaciones ahora usan CIDv1 de forma predeterminada.

3.6.3.4 NFT y el Hash Criptográfico de IPFS

Las cadenas de bloques son fuentes públicas e inmutables de verdad y el token digital que una persona ahora posee le dice al mundo quién lo creó (minter), cuándo lo crearon (altura de bloque), y el viaje transaccional que hizo antes de terminar en su billetera (transferencias). También le dice al mundo cómo se ve ese token (tokenURI o metadatos) (figura 3.6).

Figura 3.6
Un NFT desplegado en la blockchain

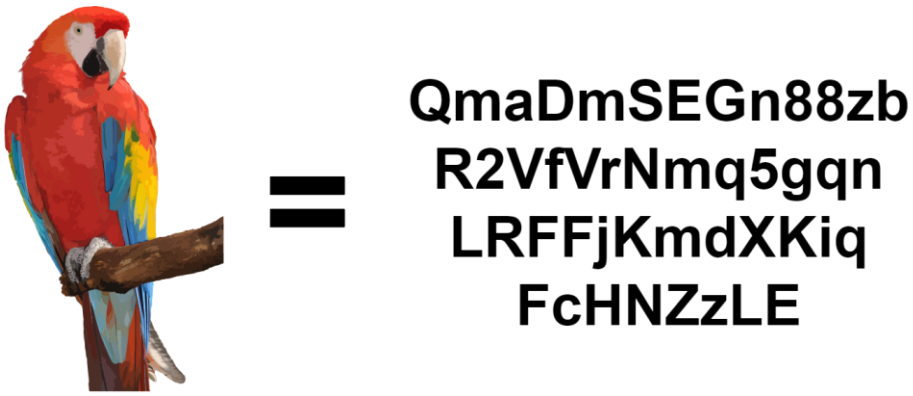


Nota, en el tokenURI podemos ver que el archivo del NFT está vinculado a una URL

La referencia del archivo de la NFT es una ruta http, el servidor de la misma podría no estar mañana, y los atributos inmutables volverse erróneos porque url ha caído. Aquí es donde entran los hashes de IPFS. Guardamos un archivo en IPFS ver la figura 3.7.

Figura 3.7

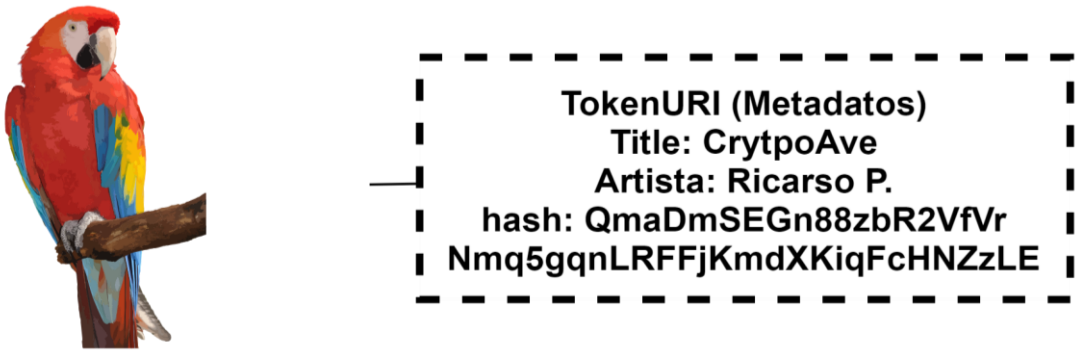
Hash de IPFS sobre la ilustración de un cryptoave



Como el hash es único para la ilustración, este es el tokenURI que se usaría para guardarla en la blockchain, de esta manera los NFT aumentan su robustez asegurando la existencia del activo mientras la blockchain existan (figura 3.8).

Figura 3.8

El TokenURI de un NFT utilizando IPFS



Nota, se verá que se cambió la ruta de la imagen, por el hash del archivo.

3.6.4 *Game Play*

A continuación se exponen dos crypto juegos que utilizan el estándar de token ERC 721 con mucha popularidad en la red Ethereum.

- **CryptoKitties**

  Gira en torno al comercio y cría de gatos virtuales. Los jugadores compran gatos nuevos del mercado del juego, o comercian y subastan gatos entre sí, todos con ethereum. Los jugadores también crían o crean nuevos gatos ya sea aparear dos gatos que poseen o aparear un gato que poseen con un gato público. Dependiendo de qué gatos se aparean, se produce un nuevo gato con diferentes propiedades, cada una con diferentes rarezas. Estas rarezas son más comparables a atributos de las cartas que se encuentran en los juegos de cartas coleccionables. Gatitos de propiedad puede servir para completar colecciones que desbloquean más recompensas, pero el objetivo principal del jugador sugerido por el sitio web del juego es para simplemente coleccionar gatos, descubrir gatos nuevos o criar e intercambiar gatos con fines de lucro.

- **Evolution Land**

  Se puede describir como un mercado de tierras virtuales de igual a igual, donde los usuarios administran parcelas virtuales de tierra en una de varias áreas únicas. Como el juego implica el comercio y el modelado de estas parcelas de tierra, con ciertas áreas de mayor demanda dada su ubicación central en el mapa u otras características deseables.

- **Axie Infinite**

  Axie Infinity es un juego de tipo Play 2 Earn, en el cual, los inversionistas entran con 3 Axies, que son los NFT del juego, con el cual pelean contra otros jugadores, ganando SLP y AXS como recompensa, existen "Scholars" que consiste, en que

un inversionista, le da su cuenta a un "becado", el cual juega por un periodo de tiempo, y se le paga un porcentaje que se establece entre el "manager" y el "becado".

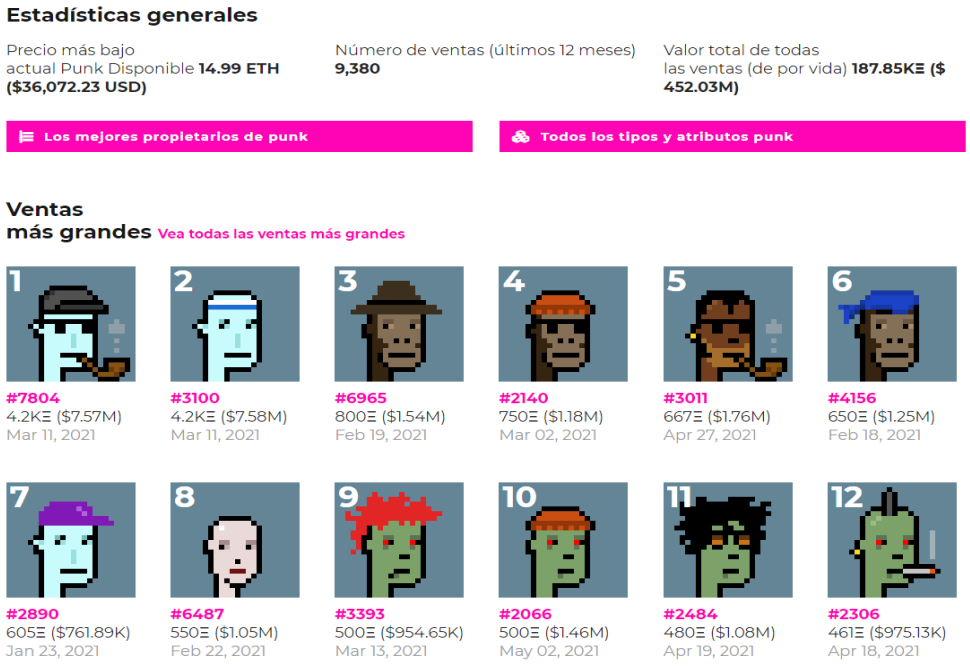
3.6.4.1 Crypto Punks

En los antecedentes para este proyecto se utilizó como referencia a Crypto punks de Larva Labs, este proyecto fue unos de los primeros de tipo NFT. El proyecto inspiraría a muchos desarrolladores y cryptoartistas. Crypto punks se haría más popular en 2021, después de que algunos crypto punks fueran vendidos por millones de dólares.

En su lanzamiento con 10000 caracteres únicos, eran gratis para su reclamo. En la actualidad están agotados y si se quiere uno, debe comprarse a alguien a través de su mercado (figura 3.9).

Figura 3.9

Mercado de ventas de crypto punks



Nota, se ve los precios de los crypto punks desde el más barato a 36 mil dólares, y el más caro de 7.57 millones de dólares, con un total de ventas de 452.03 millones de dólares. Tomada de Larba Labs, 2021 (https://www.larvalabs.com/cryptopunks)

De crypto punks podemos concluir que un activo digital puede elevar su valor, porque los usuarios le dan valor, y este aumenta en cuanto más rareza tenga, y que sigue aumentando al pasar el tiempo, todo posible por la inmutabilidad y transparencia de blockchain.

3.7 Metodología

De acuerdo con Pressman la metodología puede caracterizarse como un conjunto de técnicas para alcanzar una meta específica de desarrollo de software. Su objetivo es dar el plan de trabajo de investigación. Para lograr los objetivos del estudio y sus posibilidades se utilizó la metodología SUM, adoptado para el desarrollo dé videojuegos.

3.7.1 Metodología SUM para Videojuegos

La metodología SUM para videojuegos tiene como objetivo desarrollar videojuegos de calidad en tiempo y costo, así como la mejora continua del proceso para incrementar su eficacia y eficiencia. Pretende obtener resultados predecibles, administrar eficientemente los recursos y riesgos del proyecto, y lograr una alta productividad del equipo de desarrollo. SUM fue concebida para que se adapte a equipos multidisciplinarios pequeños (de tres a siete integrantes que trabajan en un mismo lugar físico o estén distribuidos), y para proyectos cortos (menores a un año de duración) con alto grado de participación del cliente.

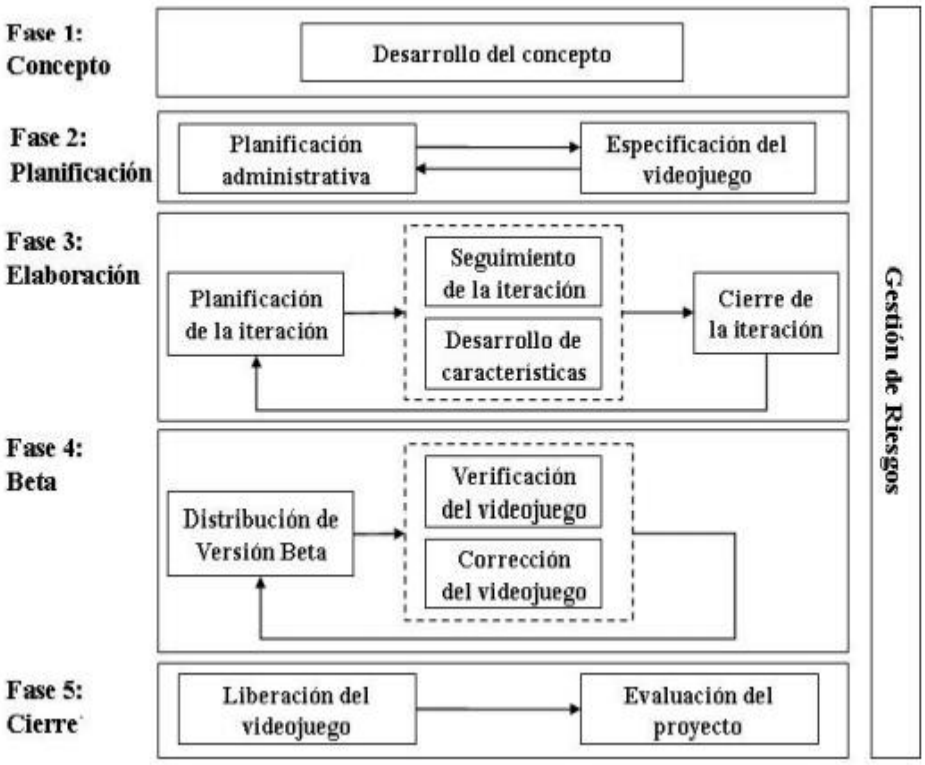
3.7.2 Ciclo de Vida

El ciclo de vida se divide en fases iterativas e incrementales que se ejecutan en forma secuencial con excepción de la fase de gestión de riesgos que se realiza durante todo el proyecto. Las cinco fases secuenciales son: concepto, planificación, elaboración, beta y cierre, como se aprecia en la Figura 3.10. Las fases de concepto, planificación y cierre se

realizan en una única iteración, mientras que elaboración y beta constan de múltiples iteraciones.

Figura 3.10

Fases del proceso metodología SUM



3.7.3 Fase 1 Desarrollo el concepto

Desarrollar el concepto del videojuego implica la realización de tres tareas para definir aspectos de negocios, de elementos de juego y técnicos. El concepto se construye a partir de ideas y propuestas de cada rol involucrado sobre los aspectos a definir. Las propuestas se refinan a través de reuniones y se analiza su factibilidad con pruebas de concepto. Estas tres tareas se realizan en paralelo, ya que se puede comenzar con cualquiera de ellas y cada una puede influenciar al resto.(Sum, 2008)

Definir aspectos del juego

- **Proponer ideas** Realizar una instancia en la cual todos puedan discutir y proponer ideas para definir la visión y características principales del juego. Es recomendable realizar bocetos para visualizar las ideas.
- **Definir la visión del juego** Describir en forma breve la experiencia que se quiere crear con el juego, que hace excitante y lo diferencia de los demás.
- **Definir el género** Identificar el género del juego, este puede estar bien definido o ser una mezcla de varios géneros conocidos. Es bueno incluir comparaciones con otros títulos del mismo género.
- **Definir el Gameplay** Definir el gameplay identificando el tipo de acciones que el jugador puede realizar durante el juego. Se recomienda incluir ejemplos.
- **Definir características** Listar las principales características del juego y detallar porque cada una es importante y como se podrán implementar. Se puede incluir desde avances técnicos hasta estilos artísticos.

Definir aspectos técnicos

- **Definir plataformas** Determinar sobre que plataformas va a funcionar el juego.
- **Definir tecnologías y herramientas** Realizar la elección de las herramientas y las tecnologías a utilizar para el desarrollo. Es importante tener en cuenta los conocimientos y capacidades del equipo.
- **Definir prototipos técnicos** En caso de requerir evaluar diferentes tecnologías se pueden realizar prototipos técnicos. Estos permiten realizar una selección informada y mejorar las estimaciones.

Definir aspectos de negocios

- **Definir modelos de negocios**
  Definir los mecanismos por los cuales el juego generará dinero.
- **Definir público objetivo**

Definir a que público está orientado el videojuego y explicar por qué puede ser de interés para este.

3.7.4 Fase 2 Planificación

La fase de planificación tiene dos objetivos principales, uno es planificar el resto de las fases del proyecto y el otro especificar las características a implementar del videojuego. Para ello se realizan dos actividades cuyos resultados componen el plan de proyecto. Estas se ejecutan en paralelo, ya que las salidas que generan dependen entre sí, por ejemplo el cronograma debe ser coherente con el tiempo estimado y para realizar las características del videojuego.

Se espera que sea una fase corta que termina cuando se tiene el acuerdo del cliente sobre los planes y características definidas. La planificación que se obtiene en esta fase es flexible, ya que en cada iteración de la fase de elaboración se puede modificar para adaptarse a los cambios y reflejar la situación actual del proyecto.(Sum, 2008)

3.7.4.1 Planificación administrativa

Esta actividad implica realizar tres tareas con el objetivo de definir diversos elementos del plan de proyecto. Se efectúan en paralelo, ya que no existe un orden de ejecución definido. Esto depende de la situación de partida al planificar, ya que, si uno o más de estos elementos están definidos previamente, los otros deben ajustarse para cumplir los requerimientos. (Sum, 2008)

- **Definir equipo de desarrollo** Seleccionar a los miembros del equipo para el resto del proyecto con base en sus habilidades y necesidades a cubrir.
- **Definir cronograma de elaboración** Definir la cantidad de iteraciones y la duración que tendrán durante la etapa de elaboración.
- **Definir presupuesto**

3.7.4.2 Especificación del videojuego

Esta actividad consta de tres tareas que se ejecutan en forma secuencial. Su propósito es describir, estimar y priorizar cada una de las características funcionales y no funcionales del videojuego.

Una característica funcional representa, en forma similar a una *User Story* de XP, una funcionalidad del videojuego desde el punto de vista del usuario final. Al ser definidas desde este punto de vista, las características son una excelente herramienta que tiene el cliente para comunicar al equipo los requisitos del videojuego y medir el avance durante todo el proyecto. Una característica no funcional representa una propiedad o cualidad que el videojuego debe presentar. Estas características suelen referir principalmente a atributos de calidad y a documentos exigidos, entre otros. (Sum, 2008)

Especificar características Se determinan y describen cuáles son las características funcionales y no funcionales del videojuego.

Estimar características Se estima el tiempo que se requiere para realizar las características del videojuego.

Priorizar características Determinar el orden en el cual deben ser desarrolladas las características del videojuego de modo de maximizar su valor.

3.7.5 Fase 3 Elaboración

El objetivo de esta fase es implementar el videojuego. Para ello se trabaja en forma iterativa e incremental para lograr una versión ejecutable del videojuego al finalizar cada iteración.

Con esta forma de trabajo se puede evaluar el avance del proyecto, lo cual permite realizar cambios a tiempo y tomar decisiones para cumplir con los plazos planificados. Además, la experiencia adquirida permite mejorar la forma de trabajo en cada iteración y aumentar

la productividad. Se espera que esta fase sea la más extensa de todo el proyecto. (Sum, 2008)

- **Planificación de la iteración** En esta actividad se crea el plan de la iteración que consta de sus objetivos, las métricas a utilizar para el seguimiento y las características a implementar. Consta de tres tareas que se realizan en forma secuencial una única vez por iteración.
    1. **Desarrollo de características** Esta actividad consta de una sola tarea en la cual se desarrollan las características planificadas para la iteración a través de la ejecución de las tareas que la componen.
    2. **Seguimiento de la iteración** Su objetivo es el de mantener la visión y el control de la iteración con base en los objetivos planteados. Consta de una única tarea que se realiza durante toda la iteración en la cual se hace el seguimiento de la misma y se toman las acciones necesarias en caso de ocurrir problemas.
    3. **Cierre de iteración** Esta actividad tiene como objetivos evaluar el estado del videojuego y lo ocurrido en el transcurso de la iteración para actualizar el plan de proyecto a la situación actual.

3.7.6 Fase 4 Beta

La fase tiene como objetivos evaluar y ajustar distintos aspectos del videojuego como por ejemplo gameplay, diversión, curva de aprendizaje y curva de dificultad, además de eliminar la mayor cantidad de errores detectados. Se trabaja en forma iterativa liberando distintas versiones del videojuego para verificar. En cada ciclo primero se planifica y distribuye la versión beta para ser verificada. Mientras esta se verifica, se envían reportes con los errores o evaluaciones realizadas. Estos reportes son analizados para ver la necesidad de realizar ajustes al videojuego. (Sum, 2008)

Se puede optar por liberar una nueva versión del videojuego para verificar una vez que se realizan los ajustes. El ciclo termina cuando se alcanza el criterio de finalización establecido en el plan de proyecto.

Distribución de versión beta Esta actividad tiene como objetivo planificar diversos aspectos de la iteración y distribuir efectivamente la versión beta para que sea verificada. Consta de dos tareas que se ejecutan en forma secuencial, planificar iteración y distribuir versión beta.

- Definir que aspectos serán tenidos en cuenta a la hora de verificar el videojuego.
- Definir las personas que participarán como verificador beta.

Verificación del Videojuego Esta actividad consta de una única tarea en la cual se verifica la versión beta del videojuego y se reportan los errores.

- Evaluar y verificar
- Reportar al equipo de desarrollo los resultados de la evaluación y los errores encontrados, según los aspectos marcados a verificar.

Corrección del videojuego La actividad tiene como objetivo la corrección del videojuego de acuerdo a los errores y evaluaciones reportadas en la verificación. Para ello se cuenta con dos tareas que se ejecutan en paralelo. En una se priorizan y determinan los cambios a realizar y en otra se realizan los cambios de acuerdo a su prioridad.

3.8.7 Fase 5 Cierre

Sus objetivos son poner a disposición del cliente la versión final del videojuego y evaluar el desarrollo del proyecto. Se compone de dos actividades que se ejecutan en forma secuencial, liberación del videojuego y evaluación del proyecto.

Capítulo IV

Marco Aplicativo

En este capítulo se expone el proceso de desarrollo del videojuego, después de los conceptos base, se expone el desarrollo del criptojuego siguiendo la metodología SUM.

4.1 Fase 1: Concepto

El concepto del videojuego debe aproximarse lo más posible a los objetivos planteados por este proyecto, se propone con el concepto tener una representación general del videojuego para su posterior desarrollo.

Idea

La idea principal para el videojuego es la colección de aves digitales con registro de propiedad, para ello utilizamos el grupo de aves que se concentran en Bolivia, con un aproximado de 1422 especies.

Visión del juego

La colección será mediante tarjetas digitales con una imagen vectorizada del ave, estas tarjetas digitales estarán conectadas a un token en la cadena de bloques, un token no fungible utilizando el estándar ERC 721 de ethereum, de esa manera agregamos valor a las tarjetas garantizando su inmutabilidad, donde el usuario o jugador podrá verificar la autenticidad y la rareza de la tarjeta.

Género

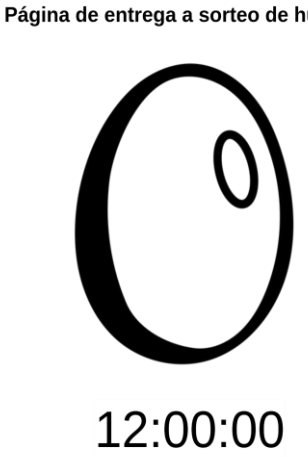
El videojuego entra en una categoría de colección y recolección unido en la cadena de bloques. Un par de juegos en el mismo género: CritptoKitties y CryptoPunks.

Gameplay

Entre las acciones principales que tendrá el jugador después de acceder mediante su billetera, es la obtención de una tarjeta gratis cada 24 horas (ver figura 4.1), solo pagando el gas por la transacción de cambio de propiedad. En cuanto a la probabilidad, varía dependiendo de la especie. Aquellas especies extintas o en peligro de extinción tendrán una probabilidad muy baja de obtención, a su vez, por su rareza solo habrá unas pocas disponibles, cada especie será dividida en cuatro posibilidades: macho, hembra, juvenil, maduro.

Figura 4.1

Ejempo de página donde entregamos huevos de ave.

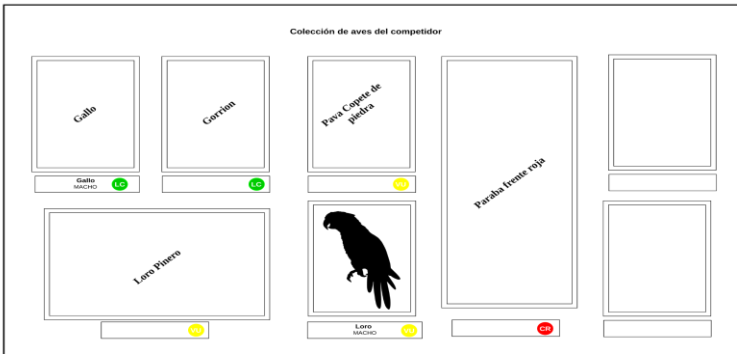


Nota, para volver a obtener un huevo se espera un tiempo de 24 horas.

Una parte importante es el álbum donde se pone la colección de aves descubiertas por parte de los usuarios (ver figura 4.2), cada álbum es público.

Figura 4.2

Ejemplo de página de un jugador



Nota, el tamaño y forma puede variar dependiendo del estado de conservación del ave.

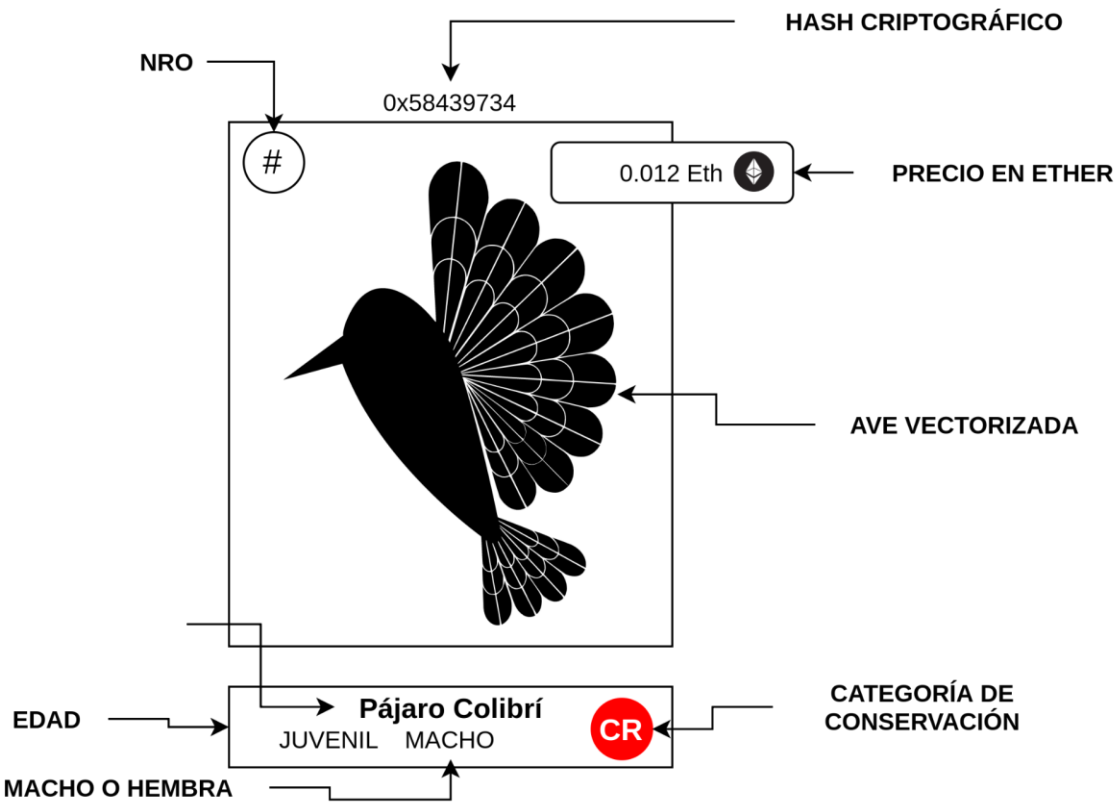
Cada tarjeta tendrá una vista de la información del ave (ver figura 4.3), donde se mostrará todo lo relacionado con la especiea la que pertenece, como ser: orden, familia, descubridor, descripción general y zona por donde habitan.

Figura 4.3

Ejemplo de información de un ave

Otra parte importante de la jugabilidad es la venta y el intercambio de las tarjetas (ver figura 4.4), se cuenta con un número limitado de cartas(ver figura 4.5), son agotables, por eso se espera que suba su valor en función del tiempo.

Figura 4.4

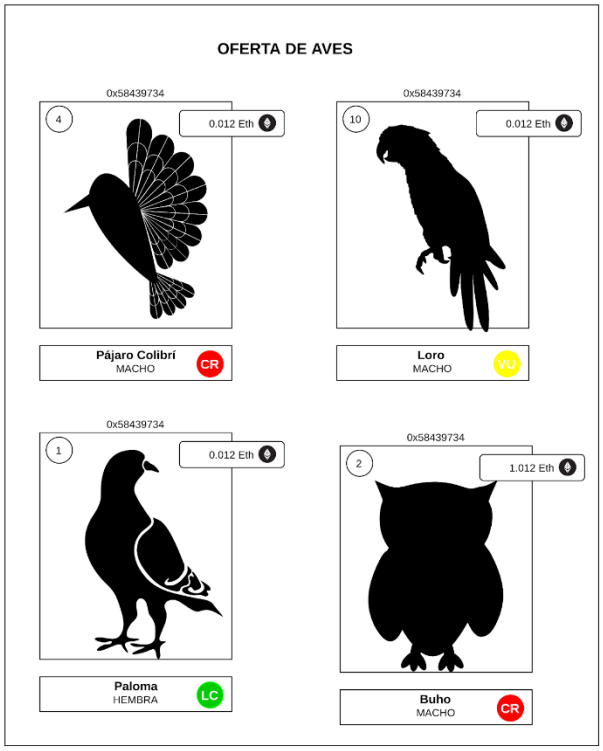


Descripción de la tarjeta de un ave

*Nota***,** en la parte superior se muestra toda la información verificable, con el hash criptográfico y el nro. En la parte inferior se muestra toda la información de rareza que tendrá la tarjeta.

Figura 4.5

Colección de aves disponibles



Nota, se lista se mezcla entre, colección de jugadores y tarjetas aún disponibles

Características

Descrito el game play, se hace un listado de las características principales para su posterior desarrollo.

1. Creación de las aves digitales vectorizadas.
2. Elaborar un estilo visual para la aplicación web del juego, diseñada en css y javascript.
3. Elaborar el servidor del juego en Nodejs, contendrá toda la funcionalidad del juego.
4. Crear los contratos inteligentes para unir las aves a blockchain.

5. Vincular la aplicación web con la red blockchain usando Metamask.

4.1.2 Aspectos Técnicos

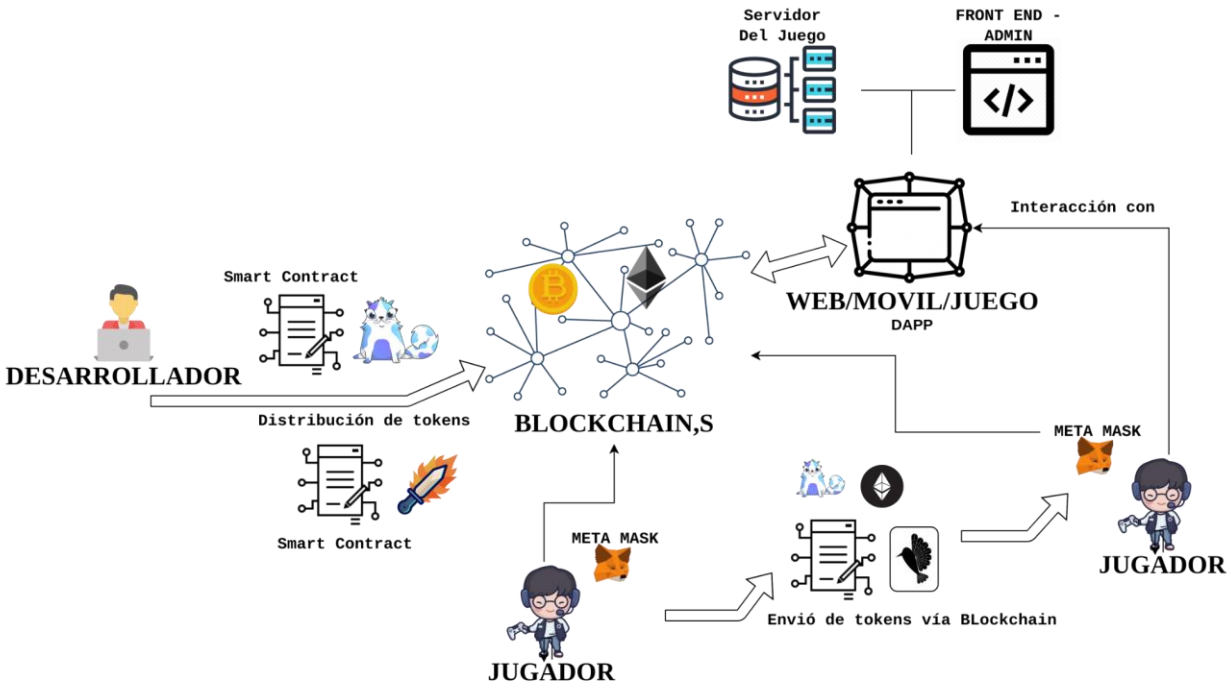
Toda la parte técnica relevante para el proyecto ya se describió en el capítulo anterior, sin embargo muchas de las herramientas que se utilizaran para el desarrollo del videojuego solo se mencionan.

Plataforma

El videojuego es multijugador online vía web (ver figura 4.6), como la mayoría de los títulos de coleccionables digitales en la blockchain.

Figura 4.6

Ilustración de las transacciones del videojuego.



Nota, el desarrollador no siempre necesita un administrador de billetera.

Herramientas

El videojuego no será enteramente descentralizado, la tecnología blockchain es utilizada para conectar tokens a las tarjetas digitales para garantizar su inmutabilidad, la aplicación web es el intermediario, y todas las mecánicas estarán necesariamente en otro servidor, el tercero entre los usuarios y la blockchain, además de ser esta webapp la parte visual y artística del videojuego.

Debido al género del videojuego sé prescinde de los motores gráficos, y de los frameworks clásicos para el desarrollo de juegos en html5, en la tabla 4.1 se lista las herramientas .

Tabla 4.1

Herramientas de desarrollo de lado del servidor

Herramientas Blockchain			
Blockchain	**Ethereum**		
	Smart Contract	**Solidity**	
	Conexión http	**web3js**	
Billetera	**Metamask**		
Entorno de desarrollo	**Truffle**		
Blockchain personal	**Ganache**		
Servidor del Juego			
Servidor	Nodejs		

	Framework MVC	Sailsjs	
Base de datos	Mysql		

Nota, se utilizan muchas más herramientas para el desarrollo, en la tabla solo muestra las principales

Tabla 4.2

Herramientas de desarrollo de lado del cliente

Frontend		
Interfaz usuario UI	Bulma	
Diseño visual		
Vectorización	Adobe Ilustrador	

Nota, para el diseño web se utiliza muchas más herramientas.

4.2 Fase 2: Planificación

Tomando como referencia las características de la fase de concepto, se estima unas cinco iteraciones la fase de elaboración, también en un estimado de una semana para cada.

Definir equipo de desarrollo

El equipo de desarrollo está a cargo de una sola persona, el autor de esta tesis.

Definir presupuesto

El presupuesto del proyecto es nulo, subir un contrato inteligente a la red principal de Ethereum conlleva un costo en ethers, también existen redes de desarrollo globales gratuitos. En cambio se optó por el desarrollo en un entorno local, que no supone gasto alguno.

Especificación del videojuego

Como recomienda la metodología, para poder especificar todas las partes del proyecto se pone al autor de esta tesis como dueño del producto y usuario administrador, entonces se procede a elaborar las historias de usuario, desde la tabla 4.3 a la 4.20.

Tabla 4.3

Historia de usuario 1

id	1	Diseño de aves digitales	
		Descripción Elaborar las aves digitales, en formato vectorial SVG	
Estimación	20		
Prioridad	Alta		Depende de 0

Nota, en la estimación se pone una X debido a la gran cantidad de aves a diseñar.

Tabla 4.4

Historia de usuario 2

id	2	Diseño de la aplicación web	
		Descripción Se desarrollará el estilo visual general de toda la aplicación web.	
Estimación	4		
Prioridad	Alta		Depende de 0

*Nota***,** esta historia es prioridad alta debido a la estética visual que exige un proyecto videojuego de colección.

Tabla 4.5

Historia de usuario 3

id	3	Diseño de un modelo de datos	
		Descripción Se elaborará la base de datos usando la herramienta de Modelo entidad-relación	
Estimación	4		
Prioridad	Alta		Depende de 0

Nota, esta historia de usuario es la base de nuestro servidor.

Tabla 4.6

Historia de usuario 4

id	4	Modo administrador	
		Descripción Se desarrollará un módulo de login administrador, para acceso a funciones de registro	
Estimación	4		
Prioridad	Alta		Depende de 0

Tabla 4.7

Historia de usuario 5

id	5	Registro de Orden	
		Descripción Se desarrollará un módulo para poder registrar las categorías de Orden de las aves	
Estimación	4		
Prioridad	Media		Depende de 4

Tabla 4.8

Historia de usuario 6

id	6	Registro de Familia	
		Descripción Se desarrollará un módulo para poder registrar las familias de aves	
Estimación	4		
Prioridad	Media		Depende de 4

Tabla 4.9

Historia de usuario 7

id	7	Registro de Descubridor	
		Descripción Se desarrollará un módulo para poder registrar al que clasifico al ave	
Estimación	4		
Prioridad	Media		Depende de 4

Tabla 4.10

Historia de usuario 8

id	8	Registro de hábitat	
		Descripción Se desarrollará un módulo para poder registrar el hábitat de las aves	
Estimación	4		
Prioridad	Alta		Depende de 4

Tabla 4.11

Historia de usuario 9

id	9	Registro de especies	
		Descripción Se desarrollará un módulo para poder registrar las aves en la base de datos del juego	
Estimación	4		
Prioridad	Alta		Depende de 4,5,6,7

Tabla 4.12

Historia de usuario 10

id	10	Creación de contratos inteligentes	
		Descripción Se desarrollará un módulo para poder registrar el ave en ethereum con el estándar erc721	
Estimación	4		
Prioridad	Alta		Depende de 0

Tabla 4.13

Historia de usuario 11

id	11	Autentificación con una billetera	
		Descripción Se desarrollará un módulo que vincule la billetera metamask de un usuario con la webapp	
Estimación	4		
Prioridad	Alta		Depende de 0

Tabla 4.14

Historia de usuario 12

id	12	Aves en etherum	
		Descripción Se desarrollará la interfaz web para módulo de asignación de cryptoAves	
Estimación	4		
Prioridad	Media		Depende de 0

Tabla 4.15

Historia de usuario 13

id	13	Obtener un ave	
		Descripción Se desarrollará un módulo para otorgar la propiedad de las aves a los jugadores	
Estimación	4		
Prioridad	Media		Depende de 3,4

Tabla 4.16

Historia de usuario 14

id	14	Colección de aves	
		Descripción Se desarrollará un módulo para mostrar las aves de los jugadores en la webapp	
Estimación	4		
Prioridad	Media		Depende de 3,4

Tabla 4.17

Historia de usuario 15

id	15	Listar aves	
		Descripción Se desarrollará un módulo para mostrar todas las aves que se encuentran alojadas en la blockchain	
Estimación	4		
Prioridad	Media		Depende de 3,4

Tabla 4.18

Historia de usuario 16

id	16	Ofertar un ave	
		Descripción Se desarrollará un módulo para poner un precio a la tarjeta	
Estimación	4		
Prioridad	Media		Depende de 3,4

Tabla 4.19

Historia de usuario 17

id	17	Intercambiar ave	
		Descripción Se desarrollará un módulo para intercambiar aves entre jugadores	
Estimación	4		

Prioridad	Media		Depende de 3,4

Tabla 4.20

Historia de usuario 18

id	4	API de Especies	
		Descripción Se desarrollará un api pública para proporcionar atributos y la imagen(svg) de cada ave	
Estimación	4		
Prioridad	Alta		Depende de 0

4.3 Fase 3: Elaboración

4.3.1 Primera Iteración

La primera iteración es donde se construirá la base del videojuego, como: la base de datos, el servidor, las aves y el aspecto visual. Se hará una evaluación al final de la misma.

Planificación de la iteración

En esta iteración se utilizan las historias iniciales y se desarrollaran cada una para empezar el proyecto, son las 1,3 y 4.

Desarrollo de característica

- Seleccionar tareas

Tabla 4.21

Extensión de característica, historia de usuario 1

Nombre	Característica	Extensión
Historia de usuario 1	Diseño de aves digitales	1. Seleccionar las aves a elaborar. 2. Diseñar las aves y vectorizarlas

Ejecutar tarea

1. Se seleccionó inicialmente siete aves a criterio del autor para la elaboración de la ilustración que posteriormente pasaran a ser NFTs. Se utilizó la herramienta de diseño adobe ilustrador, la fotografías que se usan como referencia fueron obtenidas en su mayoría de Ebirds, un portal web de ornitología para amantes de las aves, y otras fueron localizadas en internet (Ebirds, 2021).

2. Empezamos por el Yal Azul, con el nombre científico de Porphyrospiza carulescens perteneciente al orden de los passeriformes (ver figura 4.7).

Figura 4.7

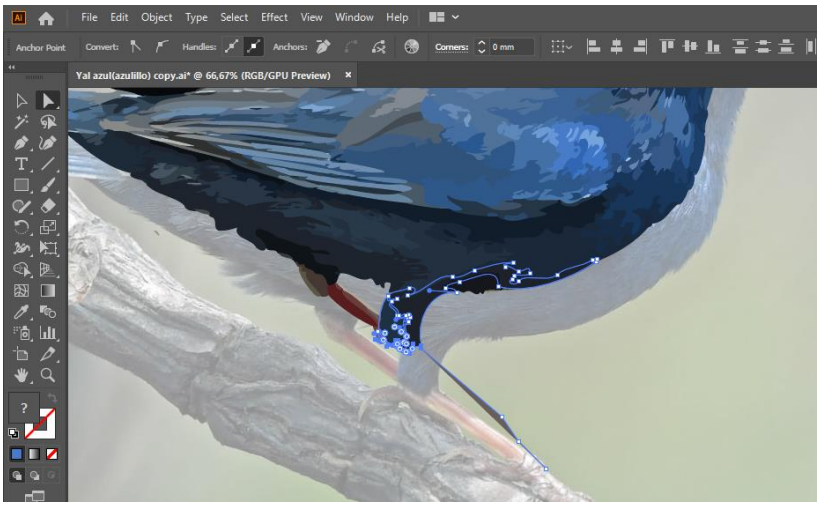
Imagen del yal azul o azulillo, adulto macho

Nota, fotografía por Tini & Jacob Wijpkema, en el cerro San Simon. Beni. Tomada de Ebirds 2021, (https://ebird.org/species/blufin1),

Continuamos llevándolo al ilustrador de adobe para su vectorización y adicionar sus colores (ver figura 4.8).

Figura 4.8

Proceso de vectorización del Yal Azul



Una vez concluida la elaboración del ave se exporta en formato svg, para su posterior uso en la tokenización (ver figura 4.9).

Figura 4.9

El yal azul vectorizada

*Nota***,** aún la ilustración no esta terminada, pasa por algunos procesos de optimización de calidad y peso.

Una vez concluida, pasamos a la siguiente ave, siguiendo el mismo proceso, entonces por simple simplificación no se muestra el diseño de las demás aves restantes.

Tabla 4.22

Extensión de característica, historia de usuario 3

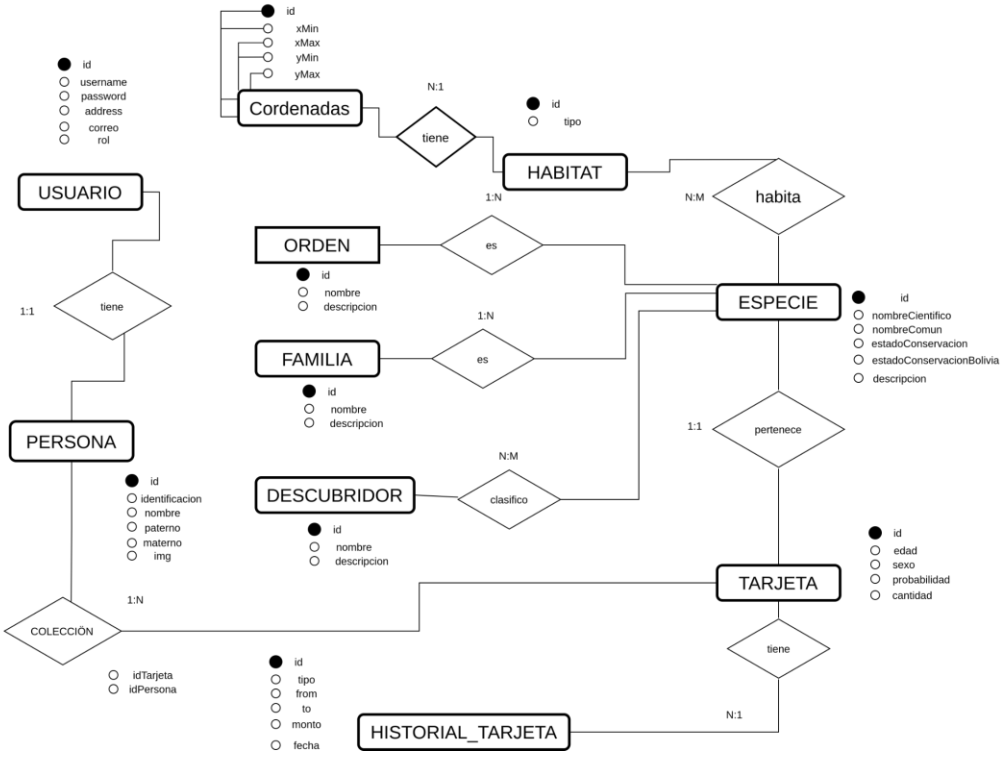
Nombre	Característica	Extensión
Historia de usuario 3	Base de datos para nuestro servidor	1. Elaborar un modelo entidad relación que contemple todo el concepto del juego e información de las aves. 2. Normalizar la base de datos

Ejecutar tarea

1. Se elaboró el modelo entidad relación (ver figura 4.10), sobre esta base nos permite construir nuestro sistema, y en las siguientes iteraciones no permite extender la base de datos.

Figura 4.10

Modelo entidad relación como modelo de datos



Nota, el diagrama es utilizado únicamente como una representación visual del almacenamiento de datos, no necesariamente será llevada a un motor de base de datos relacional.

Tabla 4.23

Extensión de característica, historia de usuario 4

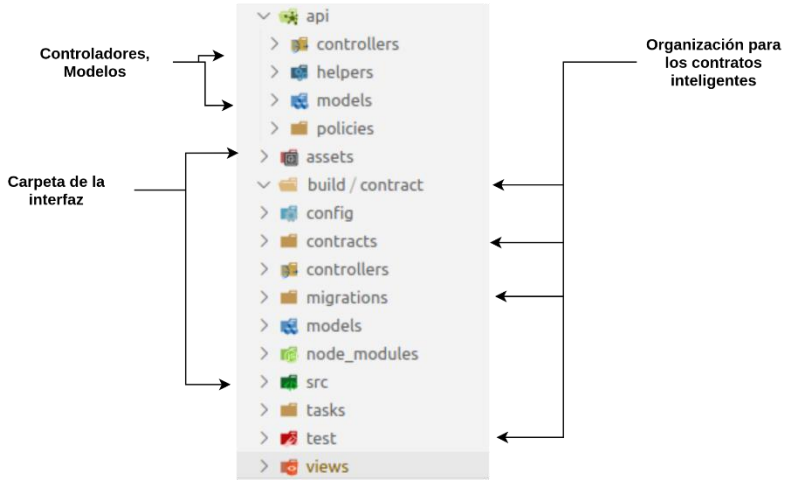
Nombre	Característica	Extensión
Historia de usuario 4	Modo Administrador	1. Crear la estructura del proyecto. 2. Adaptar el ORM de sails “waterline” para crear modelos de base de datos. 3. Elaborar el *Controller* del administrador 4. Desarrollar el formulario html para el acceso

Ejecutar tarea

1. Para la creación de la estructura del proyecto se toma en cuenta todos los aspectos del sistema, desde la interfaz web, el servidor, test de los métodos del servidor. Incluir también la estructura para los contratos inteligentes y sus pruebas (ver figura 4.11).

Figura 4.11

Estructura de carpetas interna del servidor



Nota, la carpeta de test se utiliza para los tests del servidor como de los contratos inteligentes.

2. Se empieza elaborando el modelo de usuario, se vuelve objeto para su manipulación en el controlador, para volver del modelo relacional a objeto usamos la técnica de Asignación objeto-relacional (ORM), esto nos los proporciona el adaptador para nodjes *waterline*, paquete incluido en nuestro *node_modules* al instalar sailsjs. Como

3. Procedemos a elaborar el controlador para nuestro administrador, usamos los paquetes npm de *passport*, *bcrypt-nodejs* y lo adaptamos a sailsjs

4. Como punto final elaboramos la vista para poder operar el sistema desde el modo administrador (ver figura 4.12).

Figura 4.12

Formulario de acceso del administrador

Nota, solo se tiene un formulario de acceso y no de registro, ya que por el momento solo tenemos un único administrador.

Cierre de iteración

Esta iteración fue relativamente corta, pero se dejan las bases para construir los siguientes módulos. Por parte de la documentación se resumió y simplifico estos módulos tradicionales en cualquier proyecto software para no ser tediosa su lectura.

Aspectos negativos

- En el diseño de las aves se tarda como un día para elaborar una sola, mucho más del tiempo esperado.

Aspectos positivos

- Se pudo encontrar una manera de elaborar las aves de una manera más productiva gracias a la opción *image trace* de ilustrador.

4.3.2 Segunda Iteración

En la segunda iteración armamos el servidor, el módulo que corresponde de las aves, toda su información y registro.

Planificación de la iteración

En esta iteración vamos a utilizar las historia 9 como base inicial para esta iteración del sistema. Para poder registrar a las diferentes especies de aves empezamos por las historias 5, 6 y 7. Necesitamos estos módulos creados para poder seguir con el módulo de registro de especies.

Desarrollo de característica

- Seleccionar tarea

Tabla 4.24

Extensión de característica, historia de usuario 5

Nombre	Característica	Extinción
Historia de usuario 5	Registro de Orden	1. Crear el controlador de Orden 2. Elaborar el módulo de registro en la interfaz web

Ejecutar tarea

1. Como hemos descrito en el modelo de datos inicial, la tabla de Orden tendrá 3 atributos: id, nombre, descripción, sobre ese modelo se elabora un controlador en el servidor, *OrdenController* (ver figura 4.13), encargado de: crear, mostrar, editar y eliminar. Todo administrado por el usuario *admin*, módulo creado previamente.

Se empieza elaborando el modelo Orden, volverlo objeto para su manipulación en el controlador, para volver del modelo relacional a objeto usamos la técnica de Asignación objeto-relacional (ORM), esto nos los proporciona el adaptador para nodjes *waterline,* paquete incluido en nuestro *node_modules* al instalar sailsjs.

Figura 4.13

Elaboración de las reglas del modelo para Orden.

Nota, ruta de archivo *api/controller/OrdenController*

Se procede a elaborar el controlador (ver figuras de 4.14 a 4.16) que interactuará con el modelo,

Figura 4.14

Elaboración del controlador de Orden / crear, en el servidor nodejs - sailsjs


```
20    crear: function (req, res) {
21        try {
22            Orden.validate('nombre', req.param('nombre'));
23
24        } catch (err) {
25            //   console.log("err.code",err.code)
26            switch (err.code) {
27                case 'E_VIOLATES_RULES':
28                    return res.json({ "Error": err.ruleViolations[0].message })
29                    break;
30                default:
31                    throw err;
32            }
33        }
34        Orden.create(req.body).fetch().exec(function (err, crearAve) {
35            if (err) res.serverError(err);
36            return res.redirect('/orden/index');
37        });
38    }
```

Nota, De la misma manera seguimos adicionando los métodos a nuestro controlador, hasta completar el CRUD.

Figura 4.15

Elaboración del controlador de Orden / update en el servidor nodejs - sailsjs

```
59
60    update: function (req, res) {
61        console.log("orden/update",req.body)
62        Orden.update(req.param('id'), req.body, function Update(err, value) {
63            if (err) {
64                return next(err);
65            }
66            // return res.redirect('/orden/mostrar/' + req.param('id'));
67            return res.redirect('/orden/index')
68        });
69    },
70
71
```

Nota, la validación ya viene dada en el método de *update.*

Figura 4.16

Elaboración del controlador de Orden / index, delete en el servidor nodejs - sailsjs

```
52    index: function (req, res, next) {
53        Orden.find().exec(function (err, list) {
54            if (err) return Error('Error');
55            return res.view({
56                result: list
57            });
58        });
59    },
60    show: function (req, res, next) {
61        Orden.findOne(req.param('id'), function Founded(err, value) {
62            if (err) {return next(err);}
63            res.view({
64                element: value
65            });
66        });
67    },
68    delete: function (req, res, next) {
69        Orden.destroy(req.param('id'), function Update(err, value) {
70            if (err) { return next(err);}
71            return res.redirect('/orden/index');
72        })
73    }
```

Nota, lo que retorna el servidor siempre son vistas, MVC es el modelo de arquitectura que se maneja con sailsjs.

2. Ahora se procede a elaborar el formulario de adición (ver figura 4.17), las vistas de edición (ver figura 4.18) y mostrar, como también el listado de los registros (ver figura 4.19), usando nuestro estilo personalizado.

Figura 4.17

Formulario de registro de Orden en la taxonomía de aves

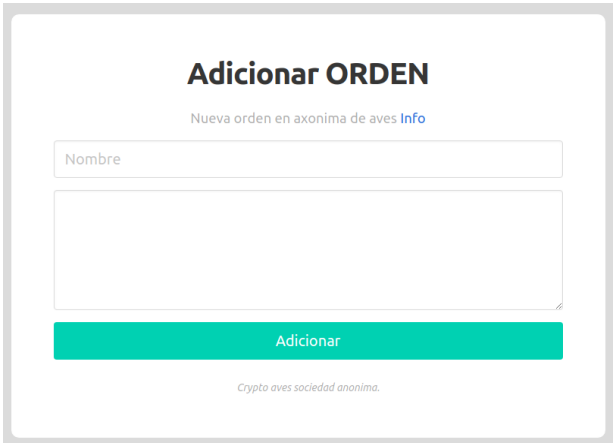
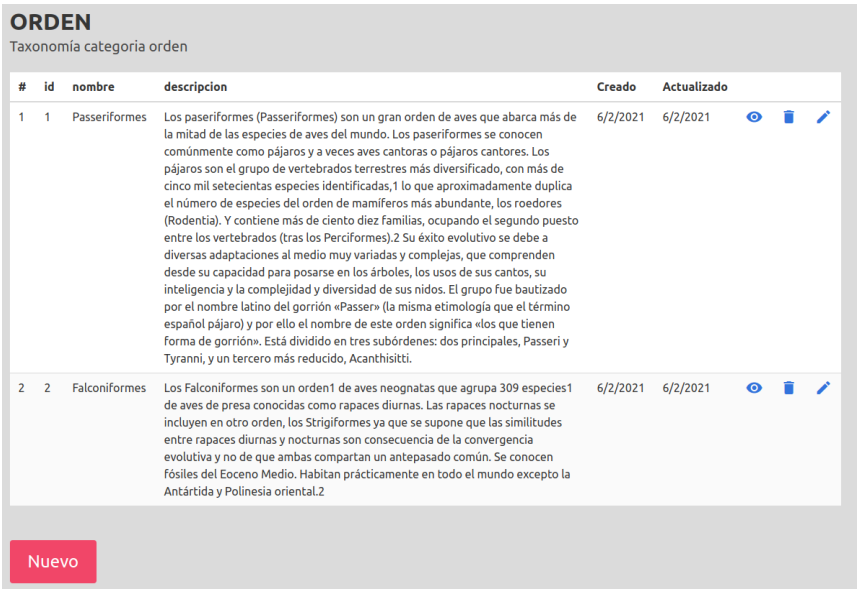


Figura 4.18

Formulario actualizar un Orden en la taxonomía de aves

Figura 4.19

Listado de los registros de Orden de Aves



Nota, el listado con dos registros de ejemplo.

Una vez concluido las tareas de la historia de usuario 5, continuamente se elaboró las historias de usuario 6 y 7 que siguen el mismo procedimiento. Por motivos de simplificación se prescinde de su documentación, se muestra a continuación los formularios de registro en la figura 4.20.

Figura 4.20

Formulario de registro de una Familia, y Descubridor respectivamente

Tabla 4.25

Extensión de característica, historia de usuario 9

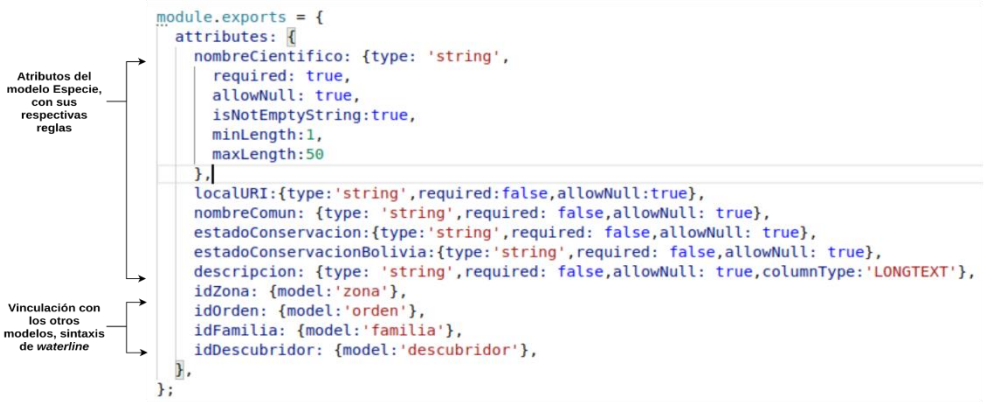
Nombre	Característica	Extinción
Historia de usuario 9	Registro de especies	1. Crear el controlador de aves 2. Elaborar el módulo de registro en la interfaz web

Ejecutar tarea

1. Se elabora el módulo de registro de aves en nuestro servidor, una vez elaborado el modelo (ver figura 4.21) seguimos con su controlador, todo elaborado de una manera similar a la historia de usuario 5.

Figura 4.21

Modelo de Especie con la especificación de sus atributos

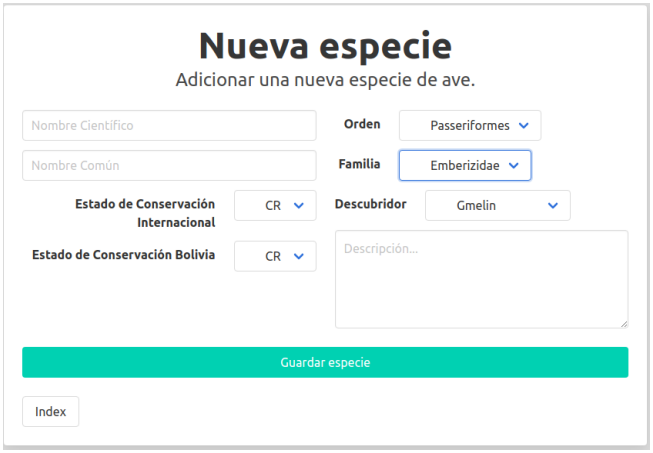


Para el controlador de especie y sus diferentes vistas continuamos de la misma manera que se hizo en *OrdenController*, por motivos de simplificación se omite el código.

2. Para la elaboración del formulario (ver figura 4.22) continuamos, necesitamos recuperar la información de los registros de Orden, Familia y Descubridor, después lo enviamos a la vista para que el usuario pueda adicionar una nueva especie.

Figura 4.22

Formulario de registro de especies.



Nota, la ubicación del archivo *views/especie/create.ejs*

Cierre de iteración

Esta iteración se encargó de armar la información de las aves en el sistema, necesaria para toda la dinámica del juego, que es principalmente mostrar a las aves.

Aspectos negativos

- El tiempo para la elaboración de cada historia de usuario sobrepaso lo estimado.
- Faltó adicionar algunos atributos en el modelo de datos propuesto.
- Falta de una fuente única información de todas las aves de Bolivia a registrar.

Aspectos Positivos

- A pesar de sobrepasar el tiempo estimado, se puede tomar en cuenta que cada historia de usuario de desarrollo en un tiempo uniforme cada uno.

Concluimos la iteración sabiendo que ahora tenemos una base general de todo el sistema, fue necesario todos los módulos desarrollados para continuar con la tercera iteración, principal del proyecto.

4.3.3 Tercera Iteración

Esta iteración es la principal del proyecto, aquí es donde se aplica toda la investigación base del marco teórico.

Planificación de la iteración.

Para esta iteración tomamos las historias de usuario 10, 11,12 y 13.

Desarrollo de característica

- Seleccionar tarea

Tabla 4.26

Extensión de característica, historia de usuario 10

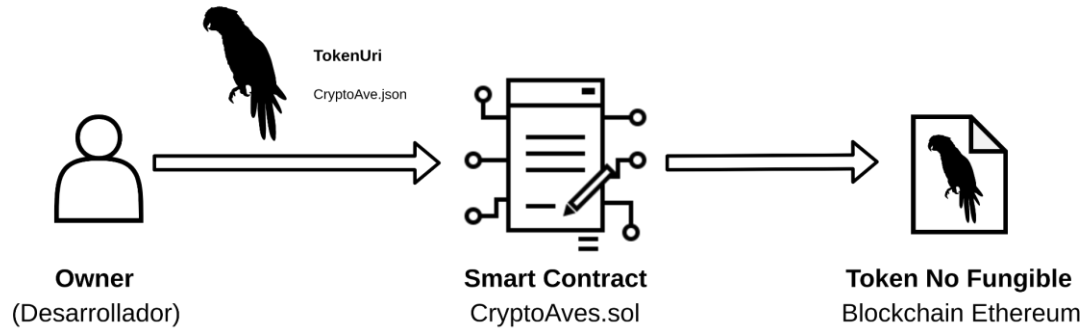
Nombre	Característica	Extinción
Historia de usuario 10	Creación de contratos inteligentes	1. Creación del contrato de registro con el estándar ERC721 2. Despliegue del contrato

Ejecutar tarea

1. El proceso de elaboración para vincular las crypto aves a un contrato inteligente (ver figura 4.23) seguirá el proceso más óptimo posible, con convenciones de código sugeridas por la misma documentación de Ethereum en las versiones actuales de Solidity y Ethereum *virtual machine*.

Figura 4.23

Proceso de tokenización de las cryptoaves en token ERC721



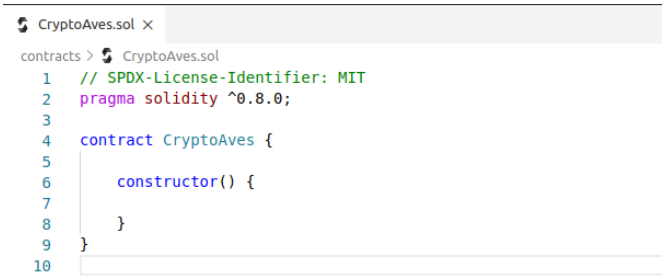
Para la elaboración del contrato inteligente de CryptoAves se apoya en la empresa OpenZeppelin, el cual se encarga de desarrollar estándares en contratos inteligentes, son

estándares mundialmente usados, minimizando el riesgo y contribuyendo a la seguridad de etherum blockchain.

Empezamos creando nuestro contrato en la carpeta de /*contracts* **CryptoAves.sol**, siguiendo la sintaxis de solidity (ver figura 4.24).

Figura 4.24

Creación de nuestro contrato.



Para que un contrato sea compatible con ERC721 requiere la siguiente interfaz

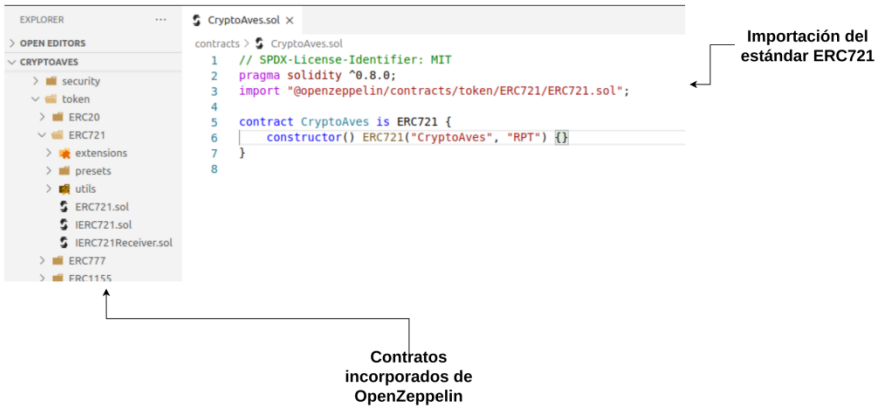
Funciones

- *balanceOf (owner)*
- *ownerOf (tokenId)*
- *safeTransferFrom (from, to, tokenId)*
- *transferFrom (from, to, tokenId)*
- *approve (to, tokenId)*
- *getApproved (tokenId)*
- *setApprovalForAll (operator, _approved)*
- *isApprovedForAll (owner, operator)*
- *safeTransferFrom (from, to, tokenId, data)*

OpenZeppelin nos proporciona este estándar escrito en solidity, lo instalamos y lo incorporamos en nuestro contrato (ver figura 4.25).

Figura 4.25

Incorporación del ERC721 estándar proporcionado por openzeppelin.

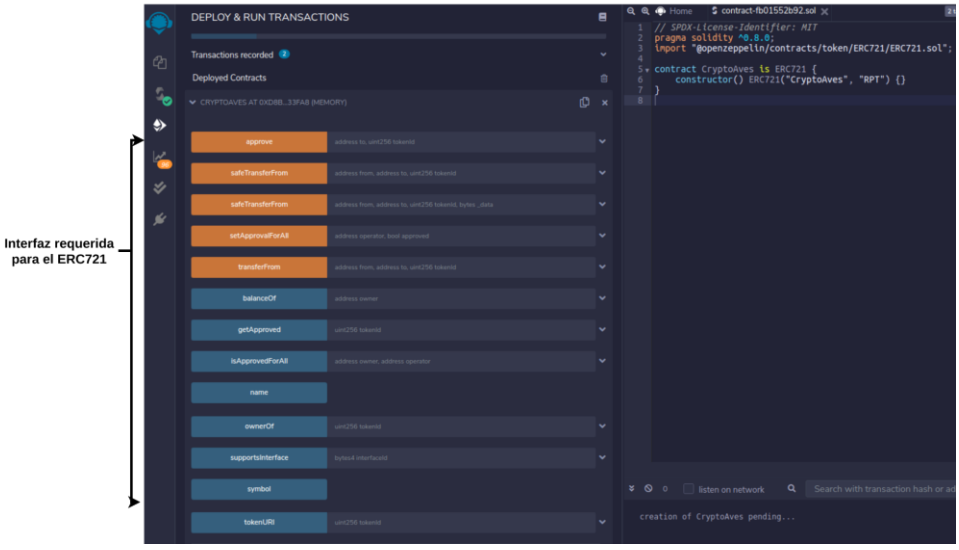


Nota, nuestro contrato hereda del contrato ERC721 obteniendo todos sus métodos.

Se usa la herramienta Remix de Ethereum para compilar el contrato, esta herramienta online nos muestra los métodos públicos del contrato que se usara después en *cryptoaves.sol (ver figura 4.26).*

Figura 4.26

El contrato adaptado al estándar ERC721.



Nota, la herramienta de Remix nos muestra las funciones de CryptoAves.sol

Finalmente creamos en contenido que tendrá el contrato, incluiremos una serie de estándares de seguridad, interfaz para manejo de token, y administrar los tokens(ver figura 4.27).

Figura 4.27

El contrato CryptoAves.sol con todas las importaciones utilizadas.

```
4   import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
5   import "@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol";
6   import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
7   import "@openzeppelin/contracts/utils/Counters.sol";
8   import "@openzeppelin/contracts/access/Ownable.sol";
9
10  contract CryptoAves is ERC721, ERC721Enumerable, ERC721URIStorage, Ownable {
```

*Nota***,** podemos ver que *solidity* soporta la herencia múltiple

Figura 4.28

Método de creación de un NFT para cryptoaves

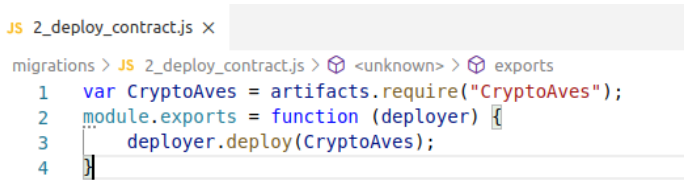
Para crear un token de cryptoaves tendremos que enviar tres argumentos: *addressJugador* es la dirección de cuenta de un usuario en el juego, *hash* representa la cadena hash

generada por IPFS de nuestra cryptoave.svg, y *metaDatos* es la dirección de todos los datos vinculados a nuestro activo (cryptoAve) en formato JSON convertida en hash igualmente por IPFS.

2. Para el despliegue usaremos *truffle* como compilador, vinculación e implementación del contrato inteligente, y *ganache* como nuestra Blockchain Ethereum personal. Antes de empezar necesitamos crear el archivo de migración de nuestro contrato (ver figura 4.29).

Figura 4.29

Archivo de migración de CryptoAves.sol



Nota, archivo en la ruta */migrations/2_deploy_contract.js*

Tenemos todo listo para el despliegue de nuestro contrato.

- Compilamos nuestro contrato con el comando

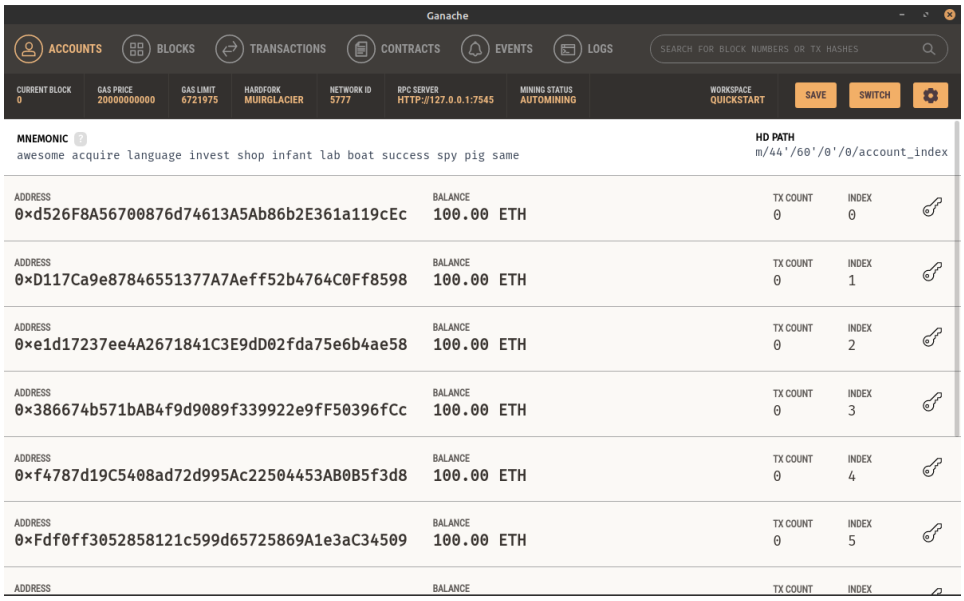
  ~$ ***truffle compile***

  Cuando compilamos, se creará el ABI del contrato, para que interactuemos con él más adelante.

- Arrancamos con la aplicación de escritorio de *Ganache (ver figura 4.30).*

Figura 4.30

Aplicación escritorio de ganache

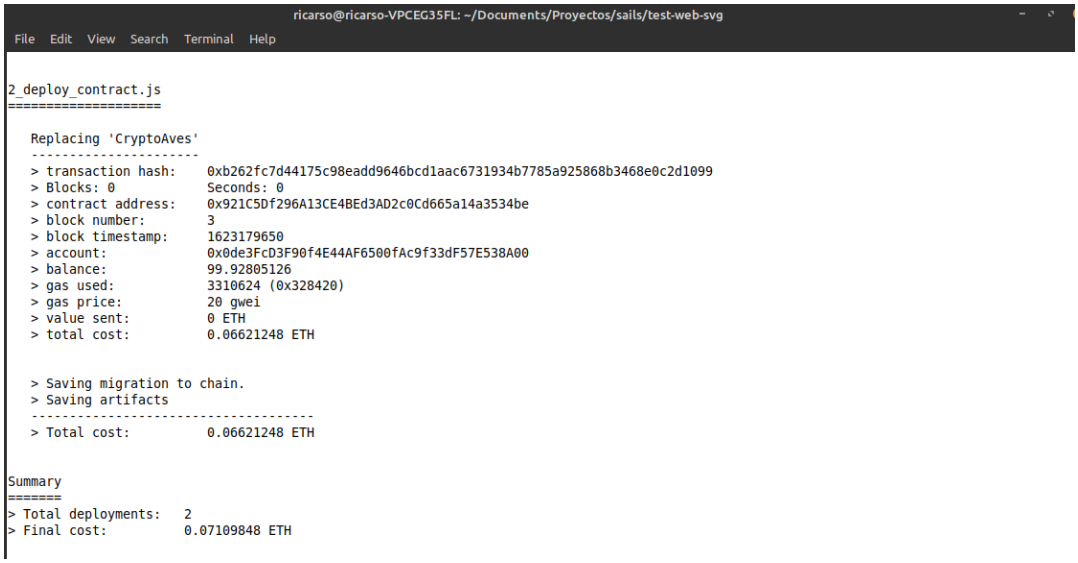


- Una vez compilado migramos nuestro contrato a ganache

  ~$ truffle migrate

Figura 4.31

Log de consola al migrar el contrato a ganache



Nota, > Contract address es la dirección del contrato en la blockchain.

Figura 4.32

La dirección de cuenta en ganache que desplegó el contrato

Nota, se puede observar el descuento de ethers como costo de migración.

Tabla 4.27

Extensión de característica, historia de usuario 11

Nombre	Característica	Extinción
Historia de usuario 11	Autentificación con una billetera	1. Incorporación de metamask a la aplicación web. 2. Interfaz web de interacción con el usuario con metamask. 3. Vincular el contrato inteligente con la aplicación web.

Ejecutar tarea

1. Meta mask es la billetera que se utilizara en la aplicación. En el navegador chrome buscamos la extensión y la añadimos. Procedemos con los pasos de configuración de inicio de metamask.

- Importación de billetera (*import wallet*)
- aceptar o rechazar el envio de analiticas
- Importar una *seed* (semilla), que nos las proporciona ganache (ver figura 4.33), y le asignamos una contraseña

Figura 4.33

Ejemplo de Mnemonic de ganache para desarrollo de pruebas

MNEMONIC

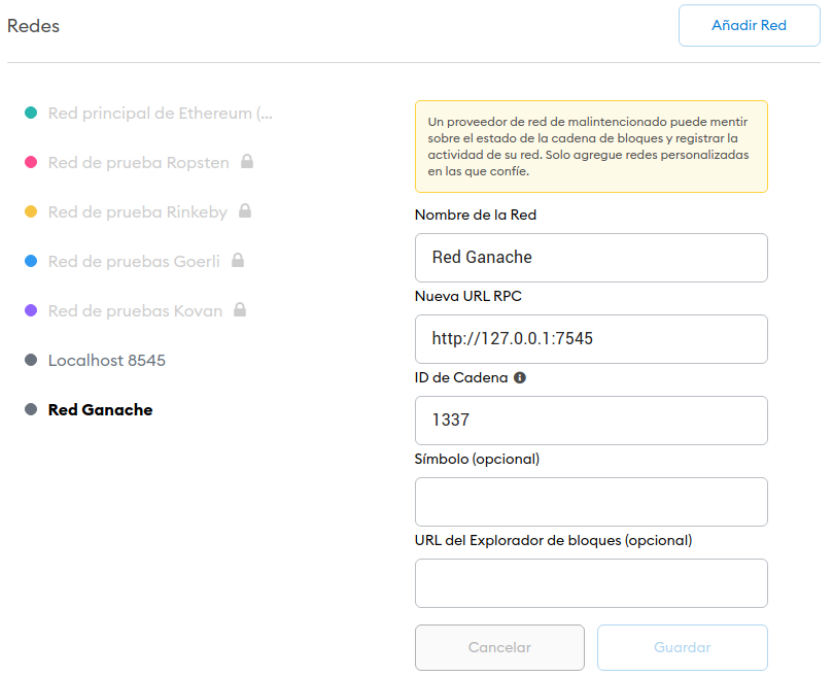
best today real guard business scheme inform sweet taste angle sheriff security

Nota, el mnemonic que ganache nos proporciona cada vez al iniciarlo

- Una vez configurado nos dirigimos a RPC personlizado para crear una nueva red.
- Añadimos una nueva red local que se vincule con *http://127.0.0.1:7545*, ruta de la red de ganache (ver figura 4.34).

Figura 4.34

Creación de la red ganache *para pruebas*

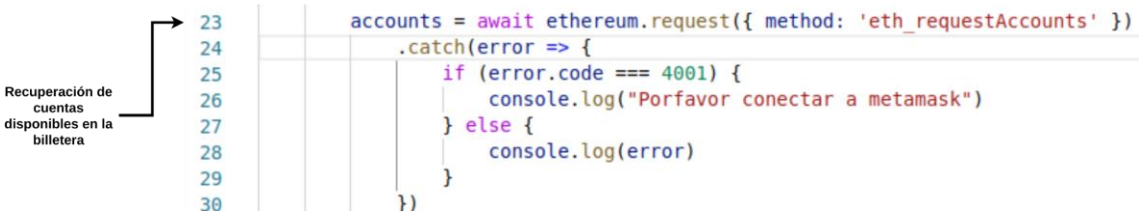


Una vez terminada la configuración de la red blockchain Ethereum local, se procede a vincular metamask con la aplicación web. Todas las configuraciones se realizan en javascript de lado del cliente, en la ruta de archivo */assets/js/configuracion.js (ver figura 4.35)*

Metamask en su documentación para administrar cuentas nos proporciona diferentes métodos para obtener la lista de cuentas actuales en la billetera del usuario, una vez obtenida la cuenta, la usaremos para hacer llamadas al contrato inteligente de cryptoaves.

Figura 4.35

Obtención de cuentas de la billetera



Nota, código simplificado para su muestra

2. Creamos un *button* para obtener las cuentas, y lo ponemos en el nav principal, después añadimos un indicador para mostrar si se esta conectado o no (ver figura 4.36 y 4.37).

Figura 4.36

Navbar principal con el boton de conectar con metamask, desconectado

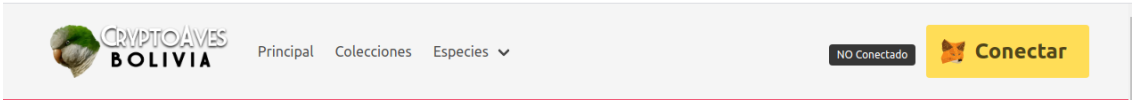


Figura 4.37

Navbar principal indicando que se está conectado

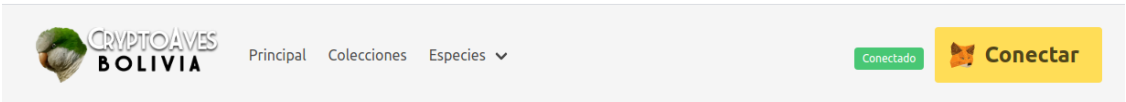
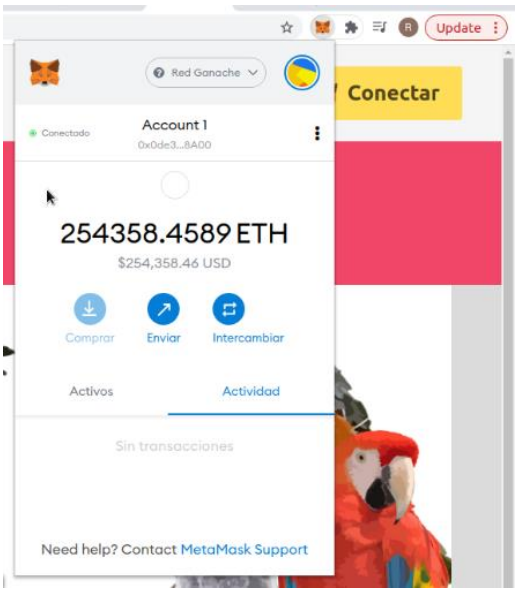


Figura 4.38

Una cuenta generada con ganache conectada con la web app.



Nota, como se puede apreciar, al vincularlo con ganache nos muestra una cantidad de ethers demasiado grande, en realidad son solo 100 eths.

3. Ahora se usará la biblioteca Web3js para interactuar con el contrato inteligente. Primero obtenemos el archivo *web3.min.js* para su incorporación en el sistema, haciendo una instancia de *Web3 (ver figura 4.39 y 4.40)* .

Figura 4.39

Instancia de web3 para manejarlo libremente

```
176  window.addEventListener('load', function () {
177      if (typeof web3 !== 'undefined') {
178          // Uso de Mist/MetaMask's provider
179          web3js = new Web3('ws://localhost:7545');
180      } else {
181          // Manejar el caso donde el usuario no tiene Metamask instalado
182          // Probablemente les muestre un mensaje pidiéndoles que instalen Metamask
183          alert("Adicionar metamask para interactuar con la web")
184      }
185      web3 = new Web3(web3js);
186      // Ahora iniciamos la aplicación y accedes a web3libremente
187      startApp();
188  });
```

Nota, en la línea 179 nos conectamos con el websocket de *ganache.*

Figura 4.40

Conexión de web3js con CryptoAves.sol

Nota, función que vincula web3 con el contrato atreves de su dirección. Código simplificado para su muestra.

Tabla 4.28

Extensión de característica, historia de usuario 12

Nombre	Característica	Extinción
Historia de usuario 12	Aves en ethereum	1. Otorgar el ave aleatoria al usuario 2. Registrar esa información en nuestro contrato.

Ejecutar tarea

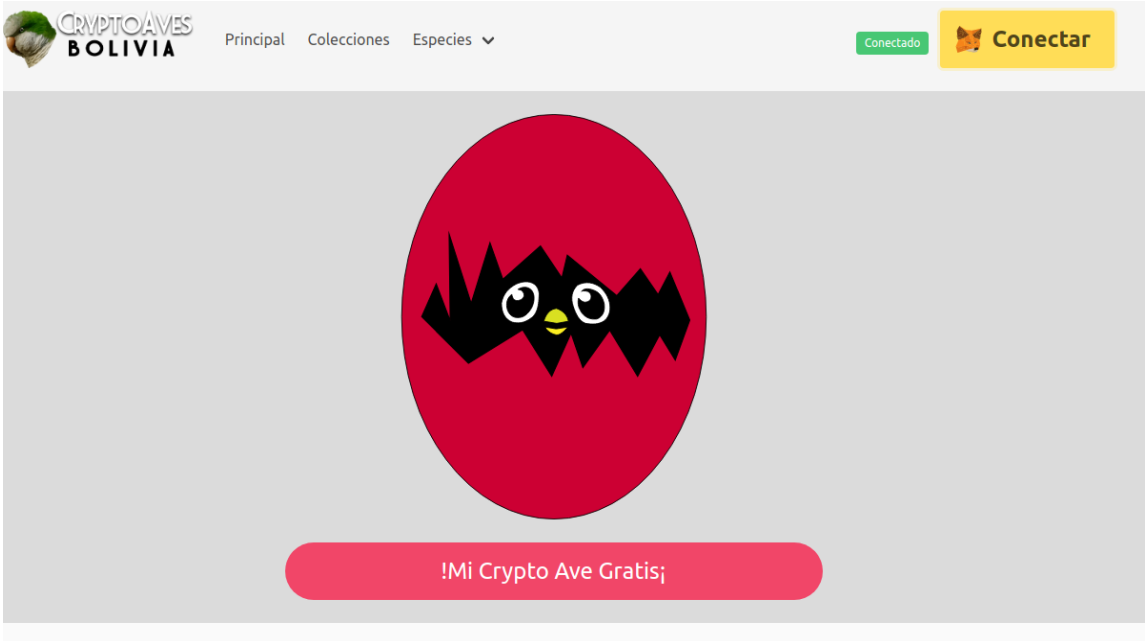
1. Se desarrolló la vista donde los usuarios podrán obtener el ave gratis cada 24 horas (ver figura 4.41), el ave viene desde el servidor, este selecciona uno aleatoriamente de todos los que aún están disponibles. El usuario obtendrá un ave y podrá registrarlo como un NFT de su propiedad (ver figura 4.42).

Para que el usuario pueda registrar al cryptoave tendrá que cumplir con los siguientes requisitos.

- Conectar una cuenta con MetaMask en la aplicación web.
- Pagar el gas necesario para la transacción.

Figura 4.41

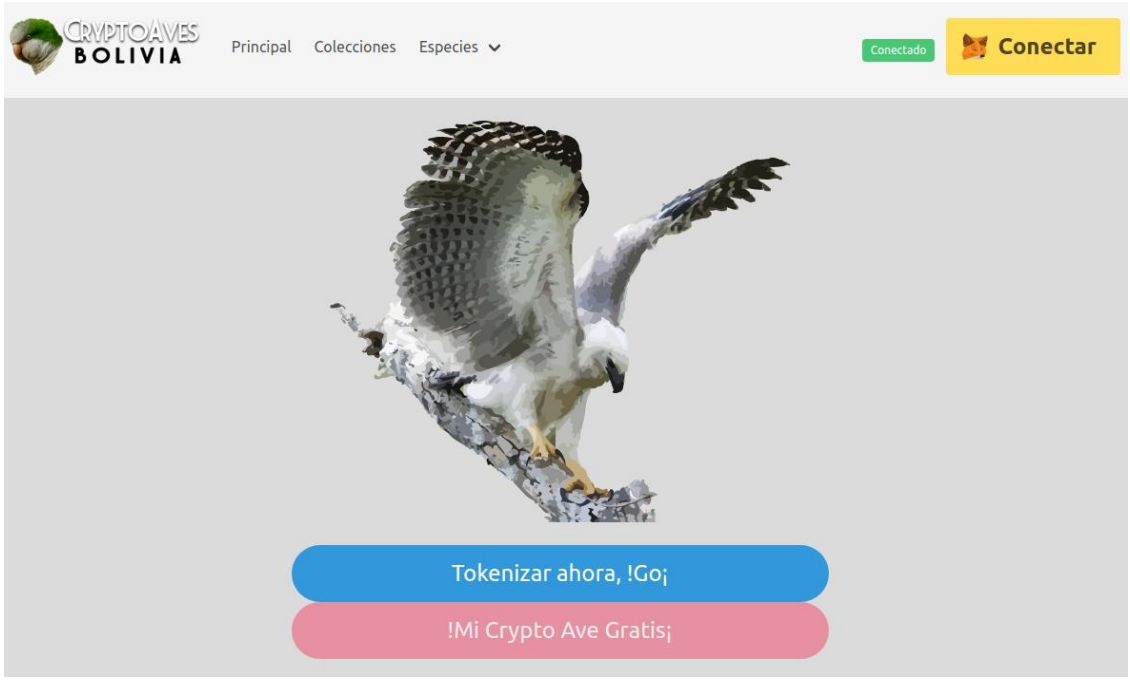
Vista de regalo de aves



Nota, página donde se regala el crypto ave de manera aleatoria

Figura 4.42

Ave otorgada de manera aleatoria



Nota, el button "Tokenizar ahora, ¡Go!" es el registro del ave

2. El proceso de registro del ave en Token sigue un estándar por recomendación de la documentación de Oppenzeppelin para los contratos ERC721, como también se aplica en las NFTs de CryptoArte de la página OpenSea.io.

Cuando se envía el crypto ave, el contrato recibe tres parámetros: *address*, *hash*, *metaDatos*. Se tiene el primer parámetro address.

Ejemplo de Address

- "**0x0de3FcD3F90f4E44AF6500fAc9f33dF57E538A00**"

Ejemplo de hash

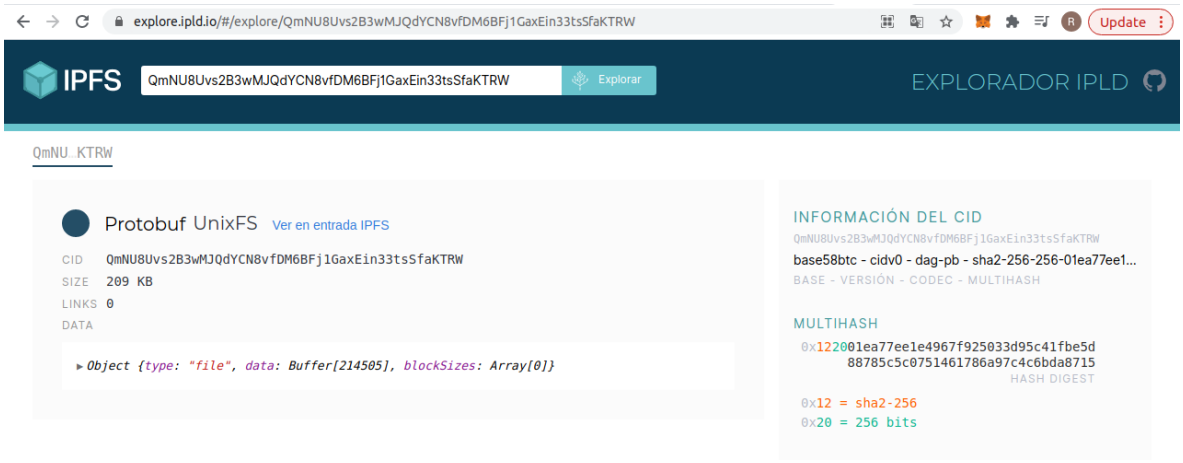
- **QmNU8Uvs2B3wMJQdYCN8vfDM6BFj1GaxEin33tsSfaKTRW**

Este hash es el identificador de contenido o CID del *ave.svg*. Usaremos el API de *Pinata* para agregar el activo a IPFS y asegurar de que permanezca anclado. También agregaremos los metadatos en JSON a IPFS para que podamos pasarlos a nuestro contrato del token.

El archivo puede buscarse desde un nodo local de ipfs, en la Figura 4.43 lo buscamos en una puerta de enlace que proporciona el equipo de IPFS, disponible en todo el mundo, y vemos el archivo en la Figura 4.44.

Figura 4.43

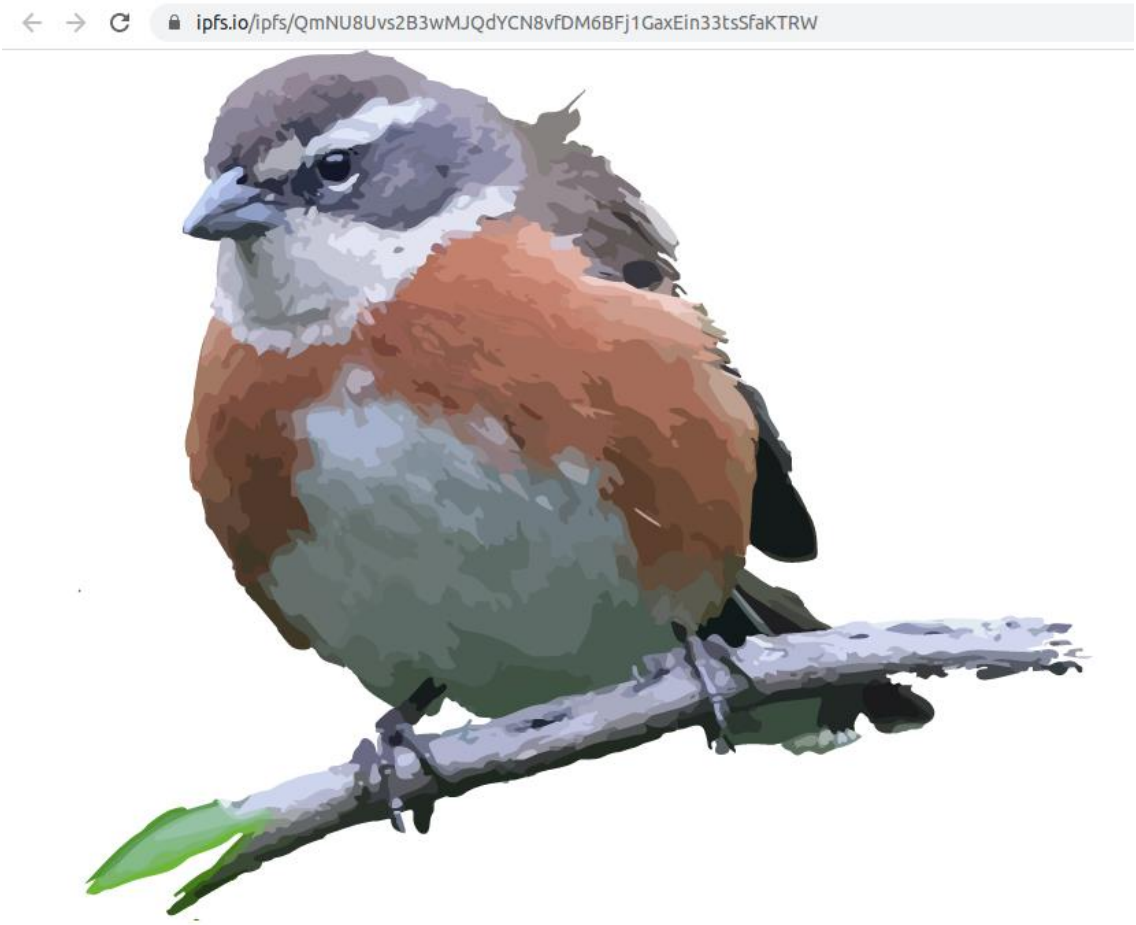
Buscando el hash en https://explore.ipld.io/



Nota, el CID utilizado es un CryptoAve de ejemplo subido a IPFS usando la puerta de enlace Pinata.

Figura 4.44

Viendo la entrada del CID en IPFS



Ahora pasamos a la construcción del parámetro *metadatos* que enviaremos. Como se vio en la creación del contrato, todas las propiedades de nuestro token no las especificamos en el contrato, como ser: nombre, edad, sexo, etc. El criterio es llevar toda esa información en un archivo JSON (ver figura 4.45), y este archivo a su vez ser llevado a Ipfs para obtener un CID nuevo representando los metadatos, y recién guardarlos en el contrato. ¿Por qué?, por la simple razón de ahorrar al máximo el gas en las transacciones, a mayor cantidad de parámetros más elevado el precio de gas, además tener los *metadatos* en un archivo JSON es una convención de ERC721.

Figura 4.45

Ejemplo de archivo.json con los metadatos

Nota, como se observa en el archivo, guardamos el hash de la ilustración

Se procede a subir este archivo a ipfs utilizando *pinata*, obtenemos su CID. Se puede verificar en *https://explore.ipld.io/*, como muestra la Figura 4.46.

Para finalizar, a *metadatos* se le agrega el protocolo de ipfs y quedo así.

- "*Ipfs://QmcXTwNytFAKsoYvdK9cogA1k75zaKuvx2Bk3RAU6APwuD*"

Figura 4.46

Viendo la entrada del CID en IPFS del archivo cryptoAves_2_monterita_boliviana.json

Ahora ya tenemos los 3 parámetros construidos, volvemos a la web app para llamar al método de creación del Token (ver figura 4.47).

Figura 4.47

Función que ejecuta la transacción, enviando address, hash, metadatos.

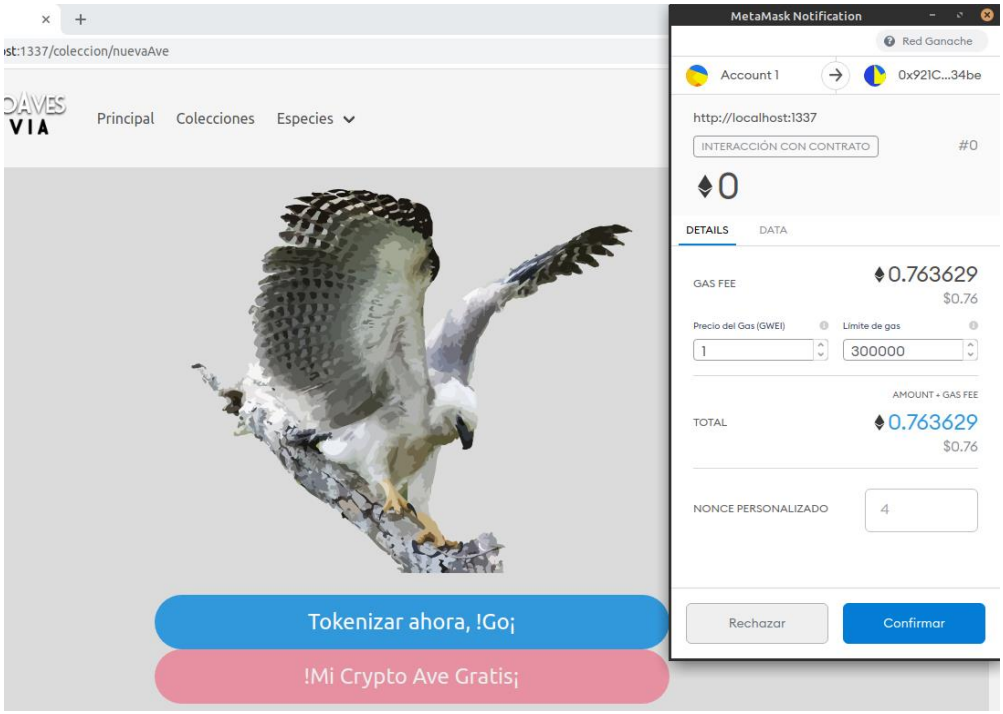
```
function otorgarItem() {
                                                    DIRECCIÓN      HASH       METADATOS
    var ipfsMetaData = "ipfs://"+aveToken.metaDatos
    const transaccion = cryptoAves.methods.otorgarItem(userAccount, aveToken.hash, ipfsMetaData)
    const transactionParameters = {
        nonce: '0x00', // Ignorado por MetaMask
        gasPrice: '0x09184e72a000', // Personalizable durante la confirmación de Metamask
        gas: '0x2710', // Personalizable durante la confirmación de Metamask
        to: '0x895345828dB6DD699504bC9869B1D01BE0560Ca5', // Dirección de contrato de CryptoAves.
        from: userAccount, // Dirección del usuario.
        value: '0x00', // Enviar dinero desde la cuenta.
        data: transaccion.encodeABI(), //llamamos al metodo.
        chainId: '0x3', // Se usa para evitar reutilizar transacciones en blockchain.
    };
    ethereum.request({
        method: 'eth_sendTransaction',
        params: [transactionParameters],
    })
        .then((txHash) => console.log(txHash))
        .catch((error) => console.error);

}
```

Nota, está es la función principal para crear los NFTs de crypto aves

Probamos este método ejecutando la función con se muestra en la Figura 4.48**.**

Figura 4.48

Confirmación de metamask para cobrar y firmar la transacción

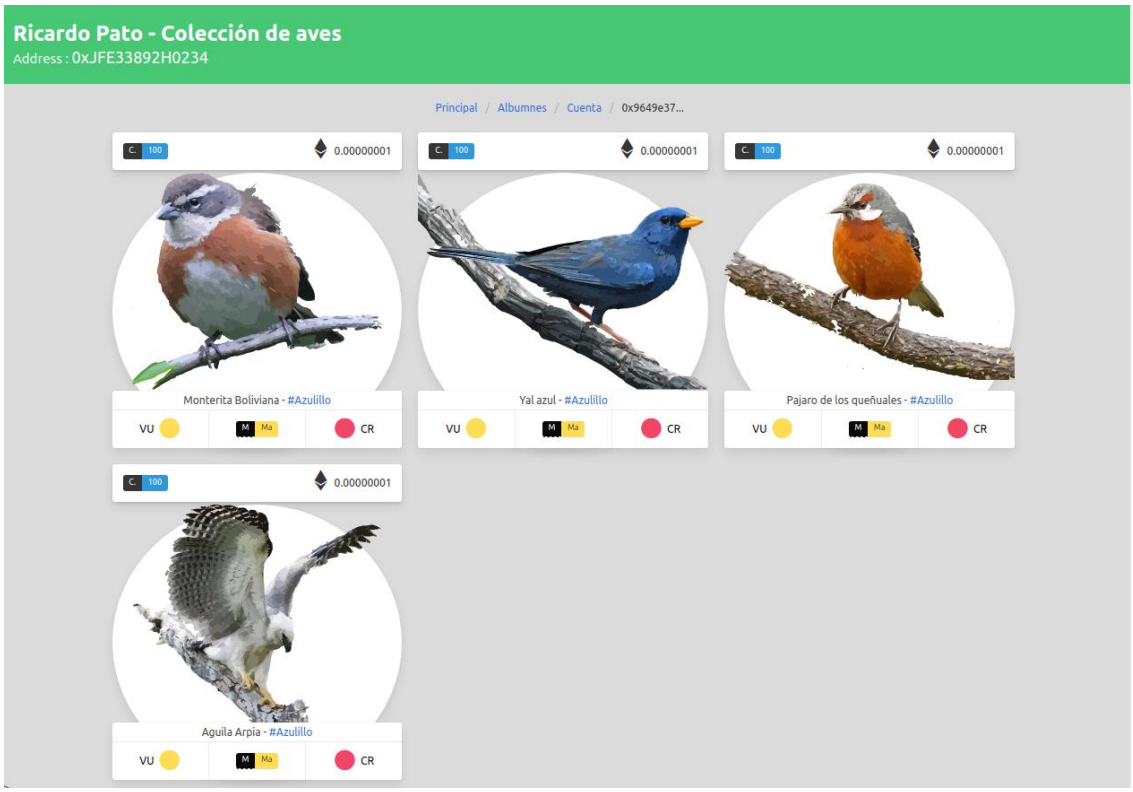


Nota, la configuración está predeterminada para que el usuario ponga el gas superior de un mínimo requerido.

De esta manera concluimos con la tokenización de los activos de nuestras aves, ahora solo queda utilizar los diferentes métodos del contrato de CryptoAves.sol para listar (ver figura 4.49), intercambiar o vender esos tokens.

Figura 4.49

Ejemplo de listado de activos de una cuenta.



Nota, podemos ver el ave Aguila Arpía ya siendo un NFT propiedad de un *address.*

Cierre de iteración

Esta iteración fue el tema central del proyecto, en el cual se creó el contrato inteligente y se adicionó el estándar ERC721 de OppenZeppelin, también vinculamos los activos a IPFS para una mayor seguridad, y finalmente usamos web3js para comunicarnos con el contrato y utilizar sus diferentes métodos.

Aspectos negativos

- El tiempo para la elaboración de cada historia de usuario sobrepaso lo estimado.
- Problemas con el manejo de errores en el despliegue del contrato y su utilización.
- Personalmente por parte del autor, encuentra escasa la documentación de web3js para su óptima utilización con metamask.
- La falta de variedad de aves creadas.

Aspectos Positivos

- El desarrollo de esta Dapp utiliza las herramientas en sus últimas versiones, como: solidity, truffle, y web3js.

Se concluye la tercera iteración del proyecto, será la última documentada por motivos de simplificación y resumen, en las siguientes iteraciones solo se desarrollará módulos restantes en vistas y acciones de los usuarios para interactuar con el contrato.

4.4 Fase Beta y Cierre

Una vez concluida la fase de elaboración se tiene una beta del videojuego disponible para ser evaluada. Este proceso de distribución de la beta se hizo con 16 usuarios evaluadores, que verificaran según aspectos marcados como: gameplay, curva de aprendizaje, dificultad, usabilidad. Por lo cual la fase beta y la fase de cierre se desarrollarán en el siguiente capítulo. También en el siguiente capítulo se propone verificar si el videojuego cumple con los objetivos del presente proyecto.

Capítulo V

Pruebas y Evaluación

Concluida la parte del desarrollo se pasa a este capítulo de pruebas y evaluación, las pruebas son una parte importante en cada proyecto al igual que el software. Se toman decisiones para probar el concepto del juego, para verificar empíricamente si cumple con el objetivo propuesto en este proyecto.

Con este fin, se reunió un grupo de 16 personas para las pruebas, 12 de ellas son expertos en desarrollo de software. Una vez se tuvo a los usuarios evaluadores, se realizó dos fases principales en la prueba del juego. Para tales fases de prueba se usó un prototipo de CryptoAves muy temprano en su creación, dado el lapso de tiempo del desarrollo del proyecto. La mayoría familiarizados con los conceptos de blockchain y NFT.

5.1 Fase de prueba I:

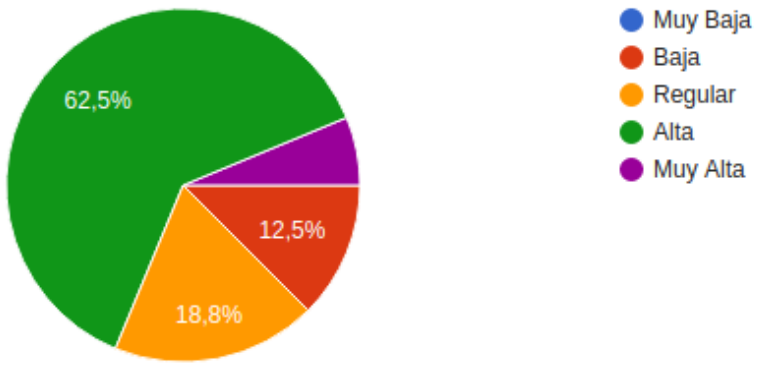
En la primera fase de prueba, se tiene como objetivo medir el conocimiento y la valoración de las aves. Para ello se preparó una encuesta para los sujetos de prueba, consistió en tres preguntas de selección múltiple.

Como uno de los objetivos del proyecto de tesis, interesa saber el valor de las aves en los usuarios. En las tablas 5.23 al 5.34, se detallan el objetivo de la pregunta, la pregunta, y los resultados.

Tabla 5.1

Tabla de resultados Fase I - pregunta 1

Encuesta Fase I

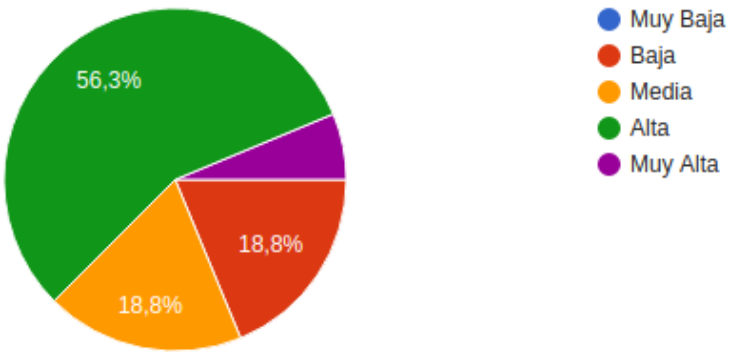
Objetivo	Determinar si el usuario está enterado del impacto ambiental de las aves.
Pregunta	¿Cuánta importancia le da al rol de las aves para el funcionamiento de los ecosistemas en Bolivia?
Gráfico	

Nota, en la pregunta se tiene con más del 68 % de usuarios que están enterados sobre el rol imprescindible de las aves para el funcionamiento de los ecosistemas.

Tabla 5.2

Tabla de resultados Fase I - pregunta 2

Encuesta Fase I	
Objetivo I	Determinar la valoración de las aves por parte del usuario.
Pregunta	¿Cuán importantes son las aves para usted?

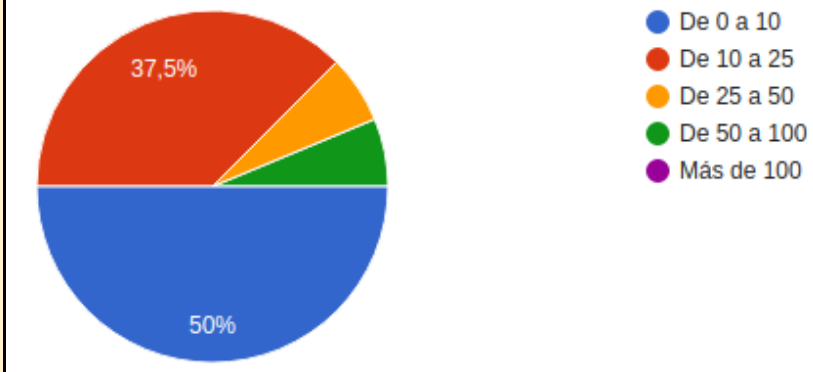
Gráfico	

Nota, la tabla demuestra que los usuarios tienen una gran valoración por las aves, los resultados de alta y muy alta superan el 60%.

Tabla 5.3

Tabla de resultados Fase I - pregunta 3

Encuesta Fase I	
Objetivo I	Medir el conocimiento de las diferentes especies de aves.
Pregunta	¿Cuántas diferentes especies de aves que habiten dentro de las fronteras de Bolivia conoce?

Gráfico	

Nota, los resultados de la tabla muestran un bajo conocimiento sobre la diversidad de las aves que existen en Bolivia, con más del 80% no superando las 25 especies conocidas

5.1.1 Conclusiones de prueba fase I

Concluiriamos que hay una valoración general de las aves, se las da la importancia esperada, pero hay una falta evidente de conocimiento en diversidad de especies por parte del grupo de usuarios, considerando las más de 1400 especies que se encuentran en Bolivia.

5.2 Fase de prueba II:

En la segunda fase de prueba tiene como objetivo principal la de verificar y evaluar la usabilidad del prototipo de videojuego de Crypto Aves Bolivia. Está fase consta de dos objetivos concretos.

Objetivo uno:

Probar la relación entre tokens y los jugadores, observar la relación entre los elementos de la cadena de bloques percibidos en el juego, centrarse por ejemplo en la adición de

tokens, ver si era complicado de entender y si los usuarios fueron incentivados al saber que eran dueños de una propiedad digital.

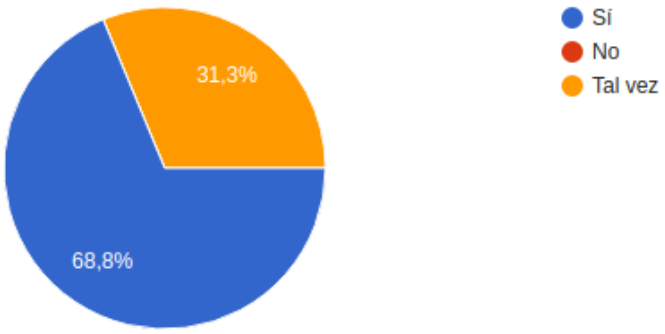
En esta fase se realizó una entrevista con el usuario, las preguntas se las hicieron personalmente, se les mostró un prototipo mínimo viable del videojuego, y se les explicó todos lo relacionado con las NFT, dando como ejemplo de videojuegos a Crypto Punks y CryptoKitties, antecedentes de este proyecto. Se les dio una dirección de cuenta en la billetera de metamask, se les pidió que registraran un ave, como también ver su colección, y ver la colección de otros jugadores, y la posibilidad de poder intercambiar.

Se preparó una encuesta, la cual consistió en tres preguntas de selección múltiple, con la ambigüedad de este objetivo, los datos obtenidos son analizados de forma cualitativa, debido a la subjetividad de las respuestas.

Tabla 5.4

Tabla de resultados Fase II - pregunta 1

Encuesta Fase II	
Objetivo	Determinar la facilidad de uso
Pregunta	¿Ha comprendido la dinámica del juego, la de coleccionar aves digitales vinculadas a la blockchain?

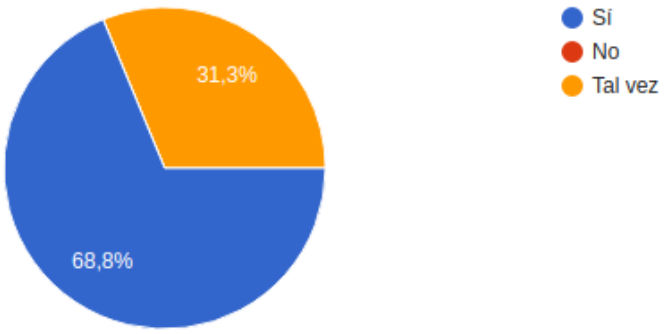
Gráfico	

Nota, para esta pregunta se les explicó el funcionamiento del videojuego se la comparo con Crypto punks y Crypto kitties, siguiendo el mismo género de juego y la mecánica con blockchain. Podemos ver en el resultado que el 68.8 % entendió y 31.3% quedaron con algunas dudas.

Tabla 5.5

Tabla de resultados Fase II - pregunta 2

Encuesta Fase II	
Objetivo I	Ver la relación entre tokens y los jugadores
Pregunta	¿Usted como jugador se sintió incentivado de querer obtener más aves para su colección?

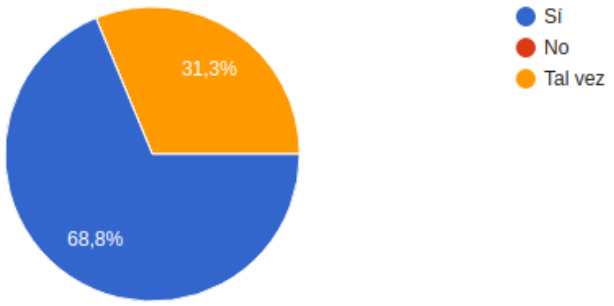
Gráfico	

Nota, al 100% de las respuestas tiene una aceptación positiva, se identificó también que estas respuestas fueron influenciadas debido a la popularidad de las NFT actualmente, también viendo la popularidad del Crypto punks en 2021, viendo el mismo potencial de crecimiento. En la prueba no se invirtió dinero real por parte de los usuarios.

Tabla 5.6

Tabla de resultados Fase II - pregunta 3

Encuesta Fase II	
Objetivo I	Determinar el aprendizaje respecto al ave
Pregunta	¿Cuándo registro al ave obtenida aleatoriamente como su propiedad usted aprendió sobre la misma?

Gráfico	

Nota, esta pregunta tuvo diferentes respuestas, todos se enteraron por lo menos del nombre del ave en cuestión, otros viendo la página de información del ave vieron más a detalle los atributos de su propiedad. Con un 68% el videojuego cumple su propósito.

Objetivo dos: Medir la jugabilidad/usabilidad del videojuego**.**

Durante todo el desarrollo se ha llamado a este proyecto de tesis como aplicación web, pero la propuesta inicial es el desarrollo de un videojuego del género colección y recolección, debería procederse a medir la jugabilidad del mismo, pero dado el género del videojuego en este caso particular, medir la usabilidad tendría el mismo enfoque que medir la jugabilidad, por lo tanto optamos por guiarnos en los criterios de usabilidad de la norma ISO 9241 -11

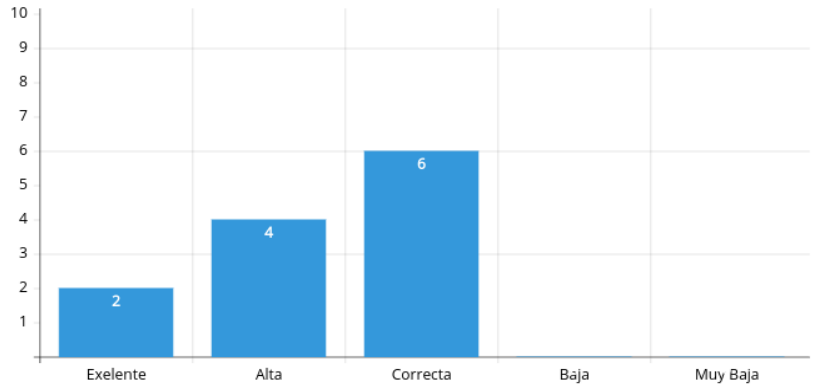
De nuestros sujetos de prueba doce son profesionales en el desarrollo de software, a los cuales se les proporcionó manejo libre del videojuego para su evaluación respecto de la usabilidad.

Navegación

¿Cuán intuitivo y fácil de recordar es el sistema a la hora de navegar por todas sus partes? (ver figura 5.1).

Figura 5.1

Resultados de encuesta, pregunta 1, sobre navegación



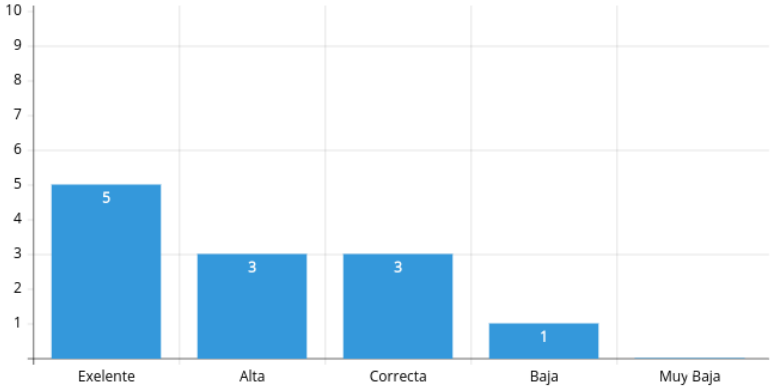
Nota, la navegación del videojuego es sencilla, la mitad lo considera correcta, y la otra mitad le dio una aceptación positiva.

Funcionalidad

¿Siendo la temática un videojuego de colección, el mismo cumple de forma explícita las características del género? (ver figura 5.2).

Figura 5.2

Resultados de encuesta, pregunta 2, sobre funcionalidad



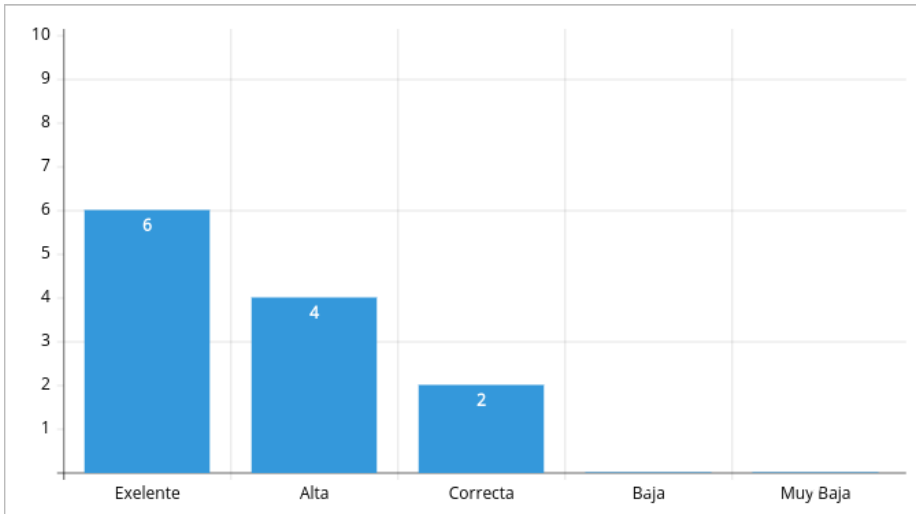
Nota, el videojuego cumple con las características del género, 8 fueron los que calificaron entre alta y excelente entendiendo que para estar vinculado a la blockchain restringe mucha jugabilidad, y otros 4 calificaron como pobre respecto al género.

Control de usuario

¿Cuánto control y libertad le permite la aplicación al usuario a la hora de explorar sus diferentes vistas? (ver figura 5.3).

Figura 5.3

Resultados de encuesta, pregunta 3, sobre control de usuario



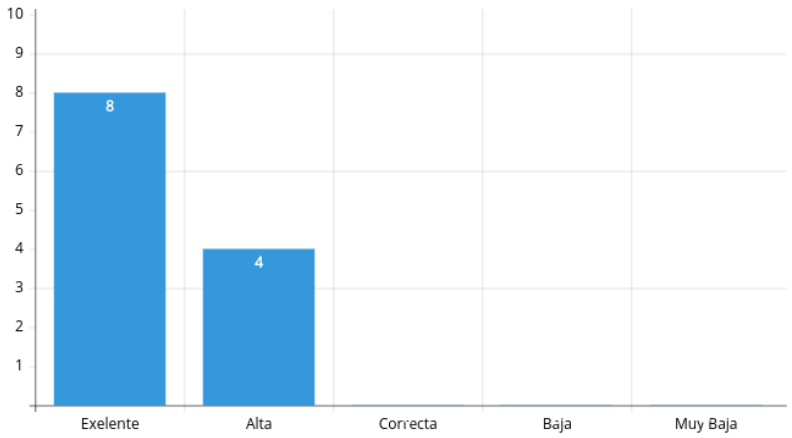
Nota, la característica de control de usuario fue tomada muy positivamente, el videojuego es simple a la hora de mostrar información.

Lenguaje y contenido

¿Los títulos, subtítulos, nombres de los elementos de la interfaz y el contenido en general, están en el lenguaje español de forma clara? (ver figura 5.4).

Figura 5.4

Resultados de encuesta, pregunta 4, sobre lenguaje y contenido



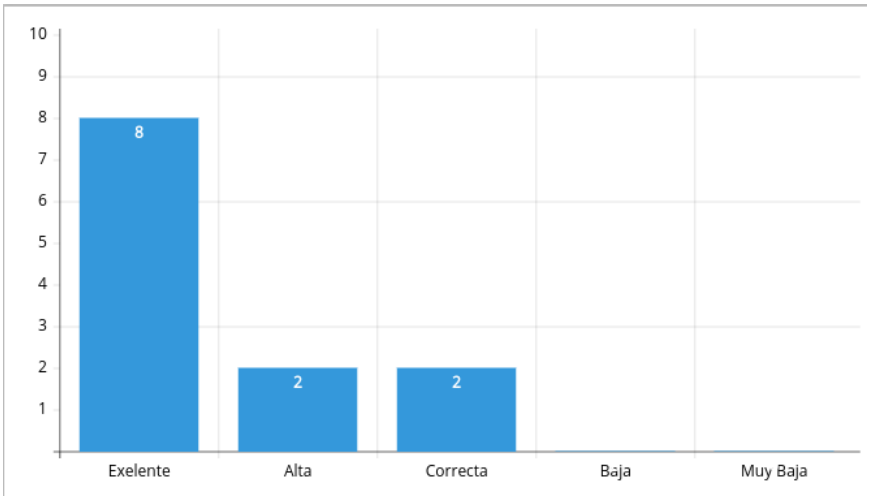
*Nota***,** los resultados de la pregunta en un 100% en alta y excelente, como se mencionó, el videojuego es simple, las instrucciones son precisas y claras.

Ayuda en línea

¿La información sobre las aves que proporciona el sistema está apoyada con páginas de terceros? (ver figura 5.5).

Figura 5.5

Resultados de encuesta, pregunta 5, sobre ayuda en línea I

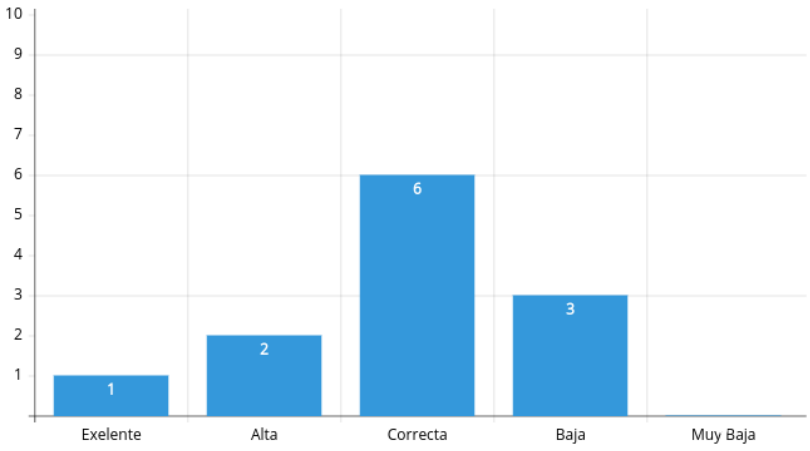


Nota, las especies, orden, familias, etc, llevan una referencia a páginas de terceros para verificar o sabes más del mundo de las aves. Las respuestas tienen una valoración óptima en este punto.

¿Cómo responde el sistema los errores, resultado de cualquier tipo de transacción? (ver figura 5.6).

Figura 5.6

Resultados de encuesta, pregunta 6, sobre ayuda en línea II



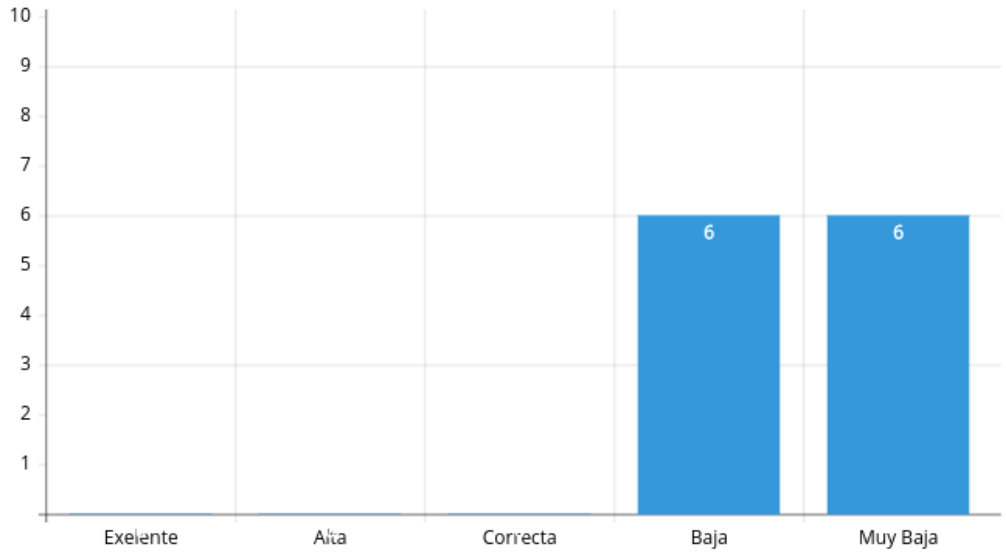
Nota, debido a que el prototipo es el mínimo viable, se le encontró poca respuesta para el manejo de errores, entre 9 respuestas fue considerada regular para abajo.

Accesibilidad

¿Cuánta calificación de accesibilidad le da al sistema con respecto a personas con discapacidades físicas sensoriales? (ver figura 5.7)

Figura 5.7

Resultados de encuesta, pregunta 7,sobre accesibilidad I

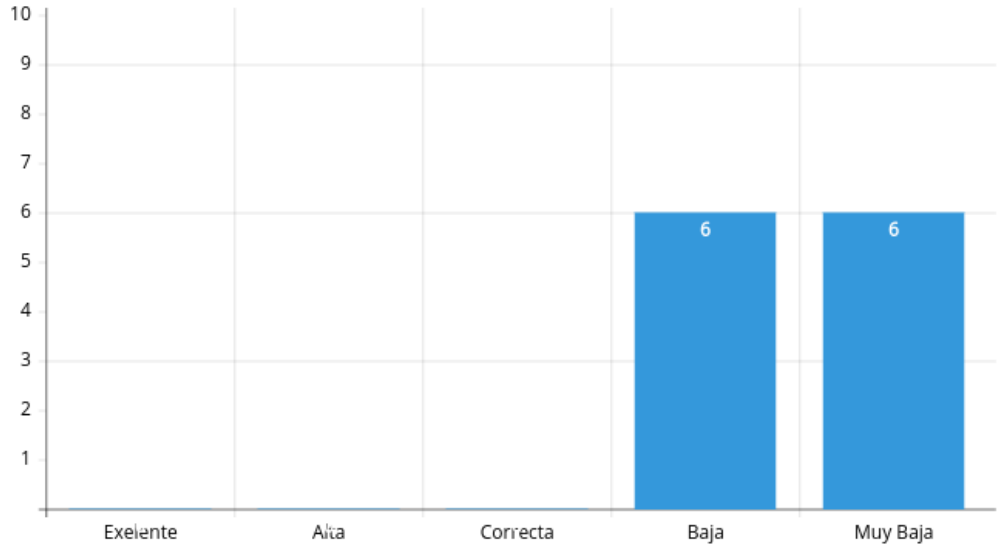


Nota, este punto no fue contemplado en el diseño del videojuego, pero como se aplica las reglas de usabilidad, también fue medido por parte de los usuarios expertos, con una calificación baja y muy baja en un 100%

¿El sistema web es responsivo adaptativo para otras plataformas? (ver figura 5.8).

Figura 5.8

Resultados de encuesta, pregunta 8, sobre accesibilidad II



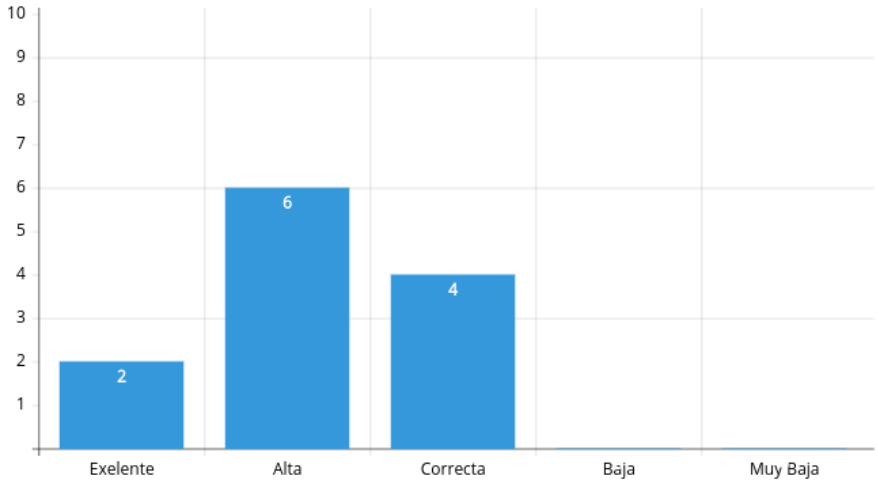
Nota, el videojuego es solo para la plataforma web en modo escritorio, las respuestas concluyen poca adaptabilidad para otras plataformas.

Prevención de errores

¿Qué tan eficiente es el sistema para guiar a los usuarios y no caigan en errores? (ver figura 5.9).

Figura 5.9

Resultados de encuesta, pregunta 9, sobre prevención de errores



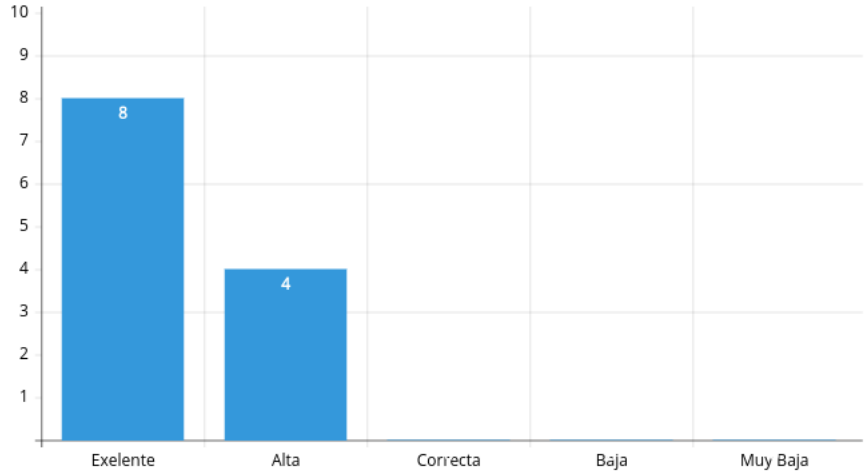
Nota, se procuró dar instrucciones precisas en la parte visual del videojuego, la mayoría de los usuarios no tuvieron problemas, pero unos 4 usuarios consideraron al sistema regular en la prevención de errores.

Coherencia

¿Cuánta calificación le da al lenguaje visual del sistema? (ver figura 5.10)

Figura 5.10

Resultados de encuesta, pregunta 10, sobre coherencia



Nota, una de las partes más relevantes del sistema es su lenguaje y estética visual, este punto tiene un 100% de aceptación positiva.

Para calcular el nivel de usabilidad del videojuego género de colección en la blockchain, se calculó el valor de ajuste de complejidad de cada atributo denominado Fi, se multiplicó la respuesta por su peso. En la tabla 4.xx podemos observar el porcentaje de respuesta para cada pregunta y su valor de ajuste.

Tabla 5.7

Tabla de evaluación de usabilidad

Nro.	% Excelente(5)	% Bueno(4)	% Normal(3)	% Bajo(2)	% Muy Bajo(1)	Fi
1	0.17	0.33	0.5	0	0	3.67
2	0.42	0.25	0.25	0.08	0	4.01
3	0.5	0.33	0.17	0	0	4.33
4	0.67	0.33	0	0	0	4.67
5	0.67	0.17	0.17	0	0	4.54
6	0.08	0.17	0.5	0.25	0	3.08
7	0	0	0	0.5	0.5	1.5
8	0	0	0	0.5	0.5	1.5
9	0.17	0.5	0.33	0	0	3.84
10	0.67	0.33	0	0	0	4.67
						35.81

Calculando la usabilidad por medio de la siguiente fórmula:

$$\text{Usabilidad} = \frac{\left(\frac{\sum \ Fi}{n} * 100\right)}{5}$$

$$\text{Usabilidad} = \frac{\left(\frac{35.81}{10} * 100\right)}{5} = 71.62\%$$

Usabilidad del 71.62%

5.3 Fase de evaluación

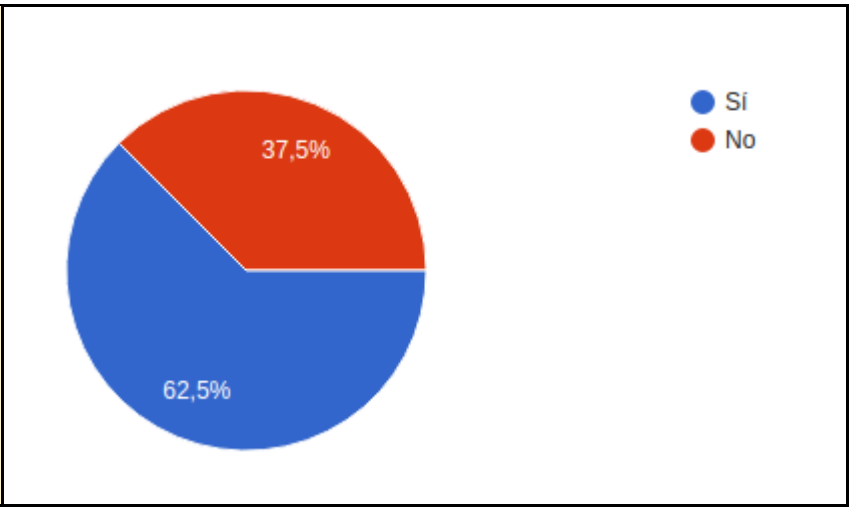
Finalmente, como fase de evaluación, se midió la valoración de las Crypto aves en NFT después de haber probado el juego.

Se preparó una encuesta, la cual consistió de una pregunta, se detalla en la tabla 5.8

Tabla 5.8

Tabla de resultados Fase III - pregunta 1

Encuesta Fase I	
Objetivo	Medir el conocimiento y la valoración de los NFT de videojuego
Pregunta	¿Cómo activo digital usted invertiría en los Token No Fungibles (NFT) de Crypto Aves?

Gráfico	

*Nota***,** después de haber probado el videojuego, los usuarios con un 62.5 % está de acuerdo en invertir, un 37.5% está con un poco de duda. Tomaron en cuenta que las aves son gratis y que solo pagarían el gas por la transacción.

Capítulo VI

Conclusiones y recomendaciones

En este capítulo final hacemos una reflexión de todas las anteriores etapas de este proyecto.

6.1 Conclusiones

Desarrollado el prototipo de cryptojuego, hechas las pruebas de usabilidad, hechas las encuestas podemos concluir los siguientes puntos.

- El objetivo general fue cumplido, ya que se logró el desarrollo del prototipo de cryptojuego para vincular aves a NFTs, donde los usuarios le otorgan un valor a estas propiedades digitales, al saber que invirtieron en ello.
- Una conclusión sobresaliente en la parte de las pruebas es la importancia del antecedente a la hora de calificar el videojuego, como ya se describió los juegos de Crypto Punks y Crypto Kitties se abren espacio en el valor subjetivo que le dan los usuarios a este tipo de videojuegos de colección con NFTs.
- También es importante llamar la atención sobre la importancia de la estética visual de las aves como activos digitales, porque depende de la calidad del mismo para ser adquiridas, valoradas incluso ser consideradas crypto arte.
- De las tablas 5.1, 5.2 de encuestas que tenían el propósito de medir el conocimiento y valoración de las aves, se obtuvo resultados en los cuales podemos concluir que la gran mayoría está consciente del rol de las aves en el funcionamiento de los ecosistemas, y valoran a las aves y al medio ambiente; pero este aprecio por estos animales voladores se contrasta con la encuesta de la tabla 5.3, donde hay poco conocimiento de las diferentes especies de aves, donde un 80% no conocen más de 25 especies que habiten en Bolivia, teniendo en cuenta que en Bolivia habitan más de de 1400 especies de aves.

- La evaluación de los expertos desarrolladores para la parte de usabilidad se concluyó con una usabilidad de la aplicación web del videojuego en un 71.62%.

6.2 Recomendaciones

Se sugieren algunas recomendaciones en base al trabajo de investigación realizado:

- El criptojuego se limita solo a adquirir e intercambiar NFTs de aves, pero se recomienda introducirlas a un mercado más grande, como Open Sea, ya que cumple con todos los estándares actuales de las ERC721 de Eethreum.
- Siendo el objetivo el desarrollo de un videojuego sobre blockchain, se identificaron varios problemas en su desarrollo: el extremo cuidado a la hora de desarrollar los contratos inteligentes, cada atributo debe ser especificado al mínimo, también la necesidad de testear cada función del contrato, porque una vez subido a la red principal no se podrá cambiar y se quedará allí eternamente.
- En la codificación de los contratos inteligentes en solidity, se vio la importancia de la versión, como son tecnologías relativamente nuevas, están en constante actualización o mejoramiento de las mismas, se sugiere utilizar siempre los estándares que nos proporciona Open Zeppelin.
- La creación de las aves digitales tiene como objetivo único su uso para este proyecto, para llevarlo a un entorno real tendría que personalizarse aún más las ilustraciones para que sea característico del videojuego, y no tan parecidos a sus fotografías reales como actualmente se manejan en este proyecto.

Referencias

Academy by Bit2me (2017), ¿Qué es una red P2P?. <https://academy.bit2me.com/que-es-una-red-p2p/> [consulta: 3 de Mayo de 2021].

Balaguer, Pedro. & Caracciolo, Juan. (2019). "Ethereum's scalability solutions for gaming". Instituto tecnológico de buenos aires, Argentina.

Balderrama, Jose E. 2009. Aves. Pp. 305-418. En: Ministerio de Medio Ambiente y Agua 2009. Libro rojo de la fauna silvestre de vertebrados de Bolivia. La Paz, Bolivia.

Benet, Juan (2020) "IPFS - Content Addressed, Versioned, P2P File System", <https://docs.ipfs.io/>.

Curran, Brian. "Blockchain games: The current state of blockchain gaming technology", August 2018. [online] <https://blockonomi.com/blockchain-games/>.

District0x Educational portal (2020), ¿Qué es Ethereum?. <https://education.district0x.io/general-topics/understanding-ethereum/what-is-ethereum//> [consulta: 12 de Mayo de 2021].

Documentación Open Zeppelin (2020), Tokens. <https://docs.openzeppelin.com/contracts/2.x/tokens> [consulta: 12 de Mayo de 2021].

Ethereum Página Oficial. (2021, 21 de junio). Ethereum Development Documentation. <https://ethereum.org/en/developers/docs/>.

EthHub portal (2020), Minería. <https://docs.ethhub.io/using-ethereum/mining///> [consulta: 12 de Mayo de 2021].

Frede Anderson, Viktor. (2019). ”Implementation of a tournament based distributed lottery on Ethereum”. Norwegian University and Technology, Noruega.

Kees Fani & Falco, Manuel & Hooglan, Pavel. (2018).”The blockchain based daily pixel art competition”. Universidad Técnica de Delf, Páises Bajos.

Lai, Victor. (2018). Crushcrypto. Introduction To Cryptography In Blockchain Technology. <https://crushcrypto.com/cryptography-in-blockchain/>.

Maillard, Oswaldo. (2012). “Estado de Conservación de las Aves en Bolivia” ,ResearchGate paper.

Min, Tian & Wang, Hanyi & Guo, Yaoze & Cai, Wei. (2019) “Blockchain Games: A Survey”, ResearchGate paper.

Munir, Sundas. & Sanam, Mirza.(2019). “Challenges and security aspects of blockchain based online multiplayer games” . School of information Techonology - University Halmtad, Suecia.

Nyyssola, Jesse. (2020) “Assessing the effects of blockchain in video games: Case IkuneRacers”. Faculty og information Technology and Electrical -University of Oulu, Finlandia.

Putra, Reza.(2020). “Sistem transaksi antar player pada game multiplayer wisata bromo menggunakan Blockchain”. Jurusan Teknik Informatika - Universitas islam negeri maulana malik ibrahim malang, Indonesia.

Rojas, R. E., Angulo, S. A., & Cabellero, E. J. (2015). AVES AMENAZADAS DEL DEPARTAMENTO DE SANTA CRUZ (N.o 1). Gobierno autónomo departamental de Santa Cruz.

Scholten, Oliver. & Jayy, Natham. & Deterding, Sebastian. & Drachen, Anders. & James, Alfred. & Zendle, David. (2019) “Ethereum Crypto-Games: Mechanics, Prevalence and Gambling Similarities”, ResearchGate paper.

SUM. (2008). SUM para Desarrollo de Videojuegos. <http://www.gemserk.com/sum> [consulta: 3 de Mayo de 2021].

Torrez Ramirez, Miguel.(2016).”Videojuego de acción RPG Kill Mosters Con agentes Npc basado en path planning y árboles de comportamiento”. Carrera de informática-Universidad Mayor de San Andres, Bolivia.