

Documento Diseño y Análisis Funcional

Crypto Card Game Marketplace

HOJA DE CONTROL

Proyecto	CCG Marketplace		
Entregable	Documento diseño y análisis funcional		
Autor	Dani Gilabert		
Versión/Edición	003	Fecha Versión	10/05/2021
Aprobado por		Fecha Aprobación	
		Nº Total de Páginas	

REGISTRO DE CAMBIOS

Versión	Motivo del Cambio	Responsable del Cambio	Fecha
v001	Versión inicial	autor	28/04/2021
v002	Revisión	autor	10/05/2021
v003	Revisión	autor	01/06/2021

INTRODUCCIÓN	3
Objetivos	4
ROLES (Actores)	6
FLUJO GENERAL DEL PROCESO	7
INTERFACES Y PANTALLAS	8
ESPECIFICACIÓN DE CASOS DE USO DEL SISTEMA	10
Listado Casos de uso del Sistema	10
Diagrama UML	11
Detalle Casos de uso del Sistema	12
ENTIDADES DEL SISTEMA	22
Smart contracts	24
ARQUITECTURA DE LA SOLUCIÓN	25
INFORMACIÓN DE DISPOSITIVOS DE INTEGRACIÓN	26

1 INTRODUCCIÓN

Los **juegos de cartas coleccionables** de mesa (Trading Card Game, también abreviado TCG) son un tipo de juego de cartas no predefinidas y existentes en gran cantidad y de variados tipos y características, que otorgan individualidad a cada carta, y con las cuales puede construirse una baraja (o mazo) dependiendo de las reglas del juego en cuestión. Entre las franquicias de este tipo más conocidas están [Magic: The Gathering](#).

Los CCG (**Crypto Card Game**), de la misma manera, son también los mismo que los TCG, pero estos últimos tienen un giro en la **blockchain**. Los jugadores son los únicos propietarios de las cartas, tokens (ERC721) no fungibles NFTs, que acumulan. Cada carta es única e irrepetible y estas, se pueden comprar, vender e intercambiar con el fin de armar un mazo poderoso.

1.1 Objetivos

El objetivo de este proyecto es desarrollar las bases de un marketplace (plataforma) donde los usuarios puedan adquirir (ser propietarios), visualizar, transferir las cartas (NFTs) a otros usuarios y poner a la venta para que otros usuarios puedan comprar. Todo ello en la blockchain de Ethereum.

Este proyecto completará la FASE 1, donde en una FASE 2 (fuera del alcance de este proyecto) se introduciría un sistema de configuración de mazos y un motor de jugabilidad, que por agilidad y costos se desarrollaría muy probablemente fuera de la blockchain.

Smart contracts:

- Definiremos una metodología para la creación de cartas. Cada carta será única e irrepetible, cada una de ellas será un token ERC721. A cada carta le asignaremos una serie de propiedades como el tipo de carta, color, maná, ataque, defensa, precio, activada/desactivada, **hash de la imagen**, etc.
- Definiremos los métodos para comprar cartas (tokens NFTs) y transferirlos a otro usuario/jugador.
- Implementaremos la lógica para que el usuario pueda vender sus cartas

Publicaremos todos los contratos en la testnet de ropsten de Ethereum.

Desarrollaremos una **dApp** para poder:

- Crear las cartas. A cada carta le asignaremos una serie de propiedades, además de un hash referenciado a una imagen. Esta imagen estará alojada en IPFS.
- Visualizar la cartas que podemos comprar y una opción para poder comprarlas.

- Visualizar la compra de cartas (en propiedad).
- Transferir una carta a otro usuario.
- Poner a la venta una carta y poder deshacer esta operación.

Desarrollaremos la dApp para 2 plataformas: web y mobile. Crearemos un apk de demostración para android.

Tecnologías:

- Solidity (SC)
- Remix
- Truffle
- Ganache
- IPFS
- Ionic/Angular
- Javascript y HTML
- Android
- Testnet de Ethereum: Ropsten

2 ROLES (Actores)

1	<i>Usuario</i>
Descripción	El usuario podrá utilizar la Dapp para comprar cartas (adquirirlas), visualizarlas, transferir cartas a otros usuarios y poner sus cartas a la venta.
Comentarios	

2	<i>Administrador</i>
Descripción	Encargado de crear las cartas o NFTs. Utilizará la Dapp para crear las cartas. Puede anular o desactivar una carta siempre que no haya sido comprada por nadie. También la podrá volver a activar.
Comentarios	<i>El administrador también tiene el rol de usuario.</i>

3	<i>Propietario</i>
Descripción	Encargado de asignar los diferentes administradores al contrato.
Comentarios	<i>Es el único que puede actualizar el contrato y destruirlo. El propietario también tiene el rol de administrador.</i>

3 FLUJO GENERAL DEL PROCESO

- 1) El propietario asigna los diferentes administradores
- 2) El administrador, crea las diferentes cartas o NFTs a través de la Dapp.
- 3) El administrador puede visualizar las cartas creadas.
- 4) El administrador puede desactivar una carta para hacerla no disponible (siempre y cuando, no sea propiedad de nadie). También la podrá volver a activar.
- 5) El usuario compra cartas a través de la Dapp.
- 6) El usuario puede visualizar las cartas compradas.
- 7) El usuario puede transferir cartas a otro usuario.
- 8) El usuario puede poner a la venta y establecer el nuevo precio de la carta. También puede deshacer esta operación.

4 INTERFACES Y PANTALLAS

Dos plataformas: Plataforma web y mobile.

Básicamente la Dapp estará formada por una pantalla de login inicial y una home. Desde la home accederemos a las diferentes pantallas a través de un menú siempre visible en todo momento. El administrador tendrá acceso a una pantalla especial para poder crear las cartas. El menú también tendrá una opción para hacer Logout. Cada pantalla tendrá una barra anclada en la parte superior que indicará el título de la pantalla actual y el acceso al menú para movernos por todas las pantallas de la Dapp.

Pantalla de Login

El usuario introduce semillas y password para identificarse y crear/importar su wallet.

- Semillas
- Password
- Botón de login
- Botón de restablecer wallet

Menú

Las opciones del menú serán las siguientes:

- Mi wallet
- Mintar carta (crear nueva carta)
- Escaparate
- Mis cartas
- Logout

Pantalla de Mi wallet

Mostrará la siguiente información:

- Address
- Balance
- ¿Es usuario administrador?

Pantalla de “Mintar carta”

El administrador de la plataforma introduce (en la pantalla de login) su semilla y password de administrador para identificarse como administrador. Tendrá acceso a una pantalla especial para crear una **nueva carta**.

Formulario nueva carta:

- Nombre
- Descripción
- Color
- Tipo
- Maná
- Ataque
- Defensa
- Precio
- Imagen

Pantalla de “Escaparate”

Visualizaremos todas las cartas que estén en venta. Serán todas aquellas que no estén desactivadas y a la venta. Representaremos las cartas como tal, con todas sus propiedades e imagen.

Cada carta mostrará el precio en Ethers y tendrá un botón de “Comprar”. Cuando el usuario compre la carta, ya no estará disponible para comprarla y pasará a manos del usuario.

Si el usuario tiene el rol de administrador, podrá activar/desactivar cualquier carta del escaparate.

Pantalla de “Mis cartas”

Visualizaremos las cartas compradas del usuario. Representaremos las cartas como tal, con todas sus propiedades e imagen.

Cada carta tendrá un botón para realizar posibles acciones:

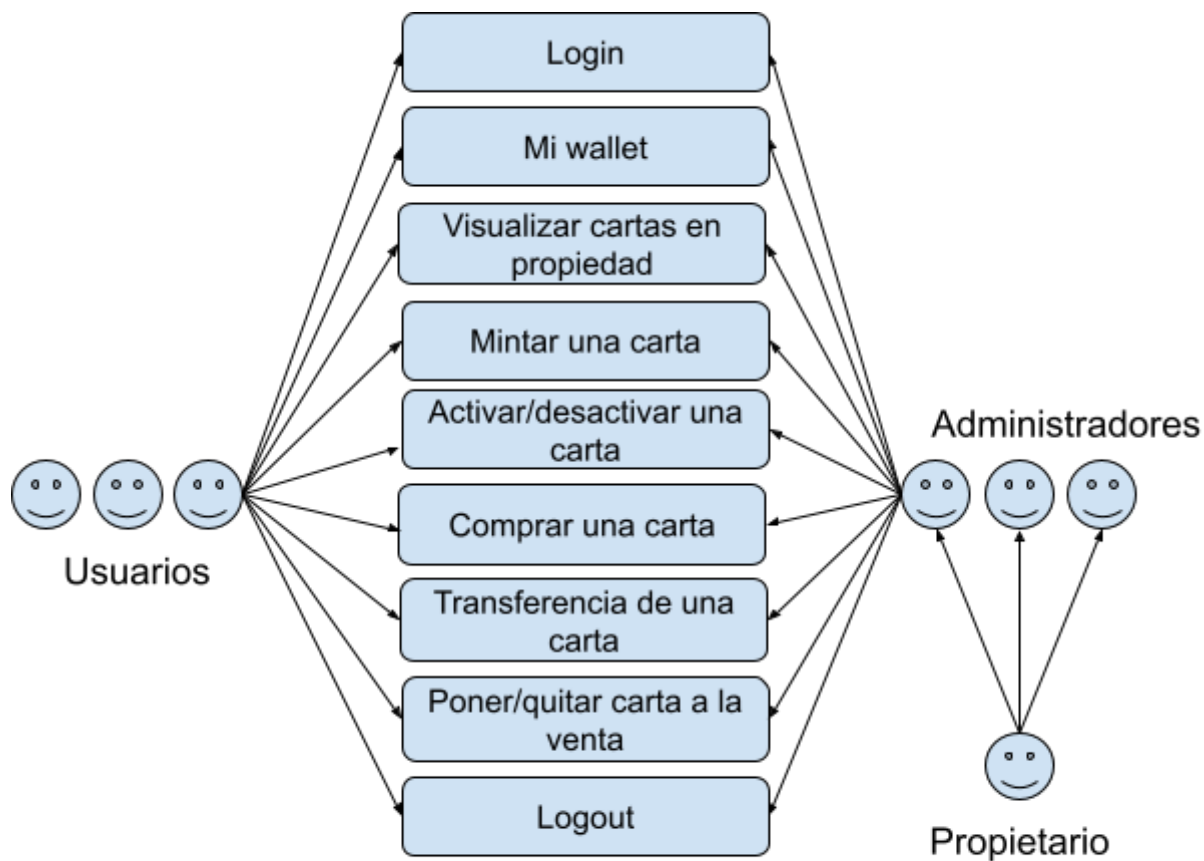
- Transferir carta: para transferir la propiedad de la carta a otro usuario.
- Poner la carta a la venta: el usuario establece el nuevo precio de la carta y la pone a la venta para que otro usuario tenga la oportunidad de comprarla. Si todavía no la ha comprado ningún usuario, se puede quitar de la venta.

5 ESPECIFICACIÓN DE CASOS DE USO DEL SISTEMA

5.1 Listado Casos de uso del Sistema

Código	Descripción	Prioridad	Dependencia
CU1	El propietario asigna a los administradores		
CU2	Login		
CU3	Mi wallet		
CU4	Visualizar cartas del usuario		
CU5	Creación de una carta		
CU6	Desactivar/activar una carta		
CU7	Compra de una carta		
CU8	Transferencia de una carta a otro usuario		
CU9	Poner/quitar la carta a/de la venta		
CU10	Logout		

5.2 Diagrama UML



5.3 Detalle Casos de uso del Sistema

CU1	<i>El propietario asigna los diferentes administradores</i>
Descripción	El propietario podrá asignar diferentes administradores.
Pantalla	No hay pantalla
Rol	Propietario
Precondición	Ser propietario
Flujo	
Flujo alternativo	
Postcondición	
Comentarios	

CU2	<i>Login</i>
Descripción	Indicaremos las credenciales para acceder a la Dapp
Pantalla	Pantalla inicial. Pantalla de Login.
Rol	Usuario y administrador
Precondición	
Flujo	<p><i>Nuevo usuario:</i></p> <ol style="list-style-type: none"> 1) Cuando se entra por primera vez, se muestran los campos de semillas y password + botón de Login. 2) Se propone al usuario una semilla por defecto generada de forma aleatoria 3) Se introduce un password 4) El usuario hace clic al botón de Login. Se encriptan las semillas junto con el password y se guarda en el LocalStorage. 5) Se muestra un recordatorio para que el usuario guarde en lugar seguro las semillas <p><i>Usuario existente:</i></p> <ol style="list-style-type: none"> 1) Se muestra un form con un password + botón de Login + botón de restablecer wallet. 2) Se introduce un password 3) El usuario hace clic al botón de Login, se desenscriptan las semillas y si es correcto, accedemos a la home. 4) El botón de restablecer wallet borrará la clave encriptada en el LocalStorage para volver a introducir una semilla nueva.
Flujo alternativo	
Postcondición	
Comentarios	

CU3	<i>Mi wallet</i>
Descripción	Nos proporciona información en tiempo real de nuestro wallet.
Pantalla	Pantalla de Mi wallet
Rol	Usuario y administrador
Precondición	Estar logeado
Flujo	<p><i>La pantalla mostrará la siguiente info:</i></p> <ul style="list-style-type: none">• <i>Address</i>• <i>Balance (saldo en Ethers)</i>
Flujo alternativo	
Postcondición	
Comentarios	

CU4	<i>Mis cartas (Visualizar cartas en propiedad)</i>
Descripción	Visualizar las cartas adquiridas por el usuario
Pantalla	Pantalla de Mis cartas
Rol	Usuario y administrador
Precondición	Estar logeado.
Flujo	<ol style="list-style-type: none"> 1) Desde la pantalla de Mis cartas, se visualizarán las cartas del usuario. 2) Cada carta estará representada por un DIV o View y en ella representaremos la imagen obtenida en IPFS y representaremos todas la propiedades con iconos y labels de texto. 3) En la parte inferior de cada carta, representaremos un botón de acción para transferir la carta i ponerla a la venta. <ol style="list-style-type: none"> a) Para transferir la carta nos pedirá la dirección del destinatario b) Para ponerla a la venta, nos pedirá el nuevo precio.
Flujo alternativo	
Postcondición	
Comentarios	

CU5	<i>Creación de una carta</i>
Descripción	El administrador crea una carta con todas sus propiedades. Le asigna una imagen, que esta se guardará en el IPFS.
Pantalla	Pantalla de Mintar cartas
Rol	Administrador
Precondición	El administrador se identifica en la pantalla de login con unas semillas preestablecidas. Estas semillas derivarán a un address que será chequeada por hardcode con la address del administrador. Accederá a la home y verá una opción especial en el menú de "Mintar cartas".
Flujo	<p><i>Formulario con los siguientes inputs:</i></p> <ul style="list-style-type: none"> ● <i>Nombre de la carta</i> ● <i>Descripción</i> ● <i>Color:</i> <ul style="list-style-type: none"> ○ <i>0: Blanco</i> ○ <i>1: Azul</i> ○ <i>2: Amarillo</i> ○ <i>3: Verde</i> ○ <i>4: Rojo</i> ○ <i>5: Negro</i> ● <i>Tipo:</i> <ul style="list-style-type: none"> ○ <i>0: Criatura</i> ○ <i>1: Hechizo</i> ○ <i>2: Reliquia</i> ● <i>Maná: 1-5</i> ● <i>Ataque: 1-5</i> ● <i>Defensa: 1-5</i> ● <i>Precio</i> ● <i>Imagen</i> <p><i>Se introducen los datos, se validan y se crea el NFT.</i></p>
Flujo alternativo	
Postcondición	La carta después de ser creada, tiene el estado de activada
Comentarios	

CU6	<i>Desactivación y Activación de una carta</i>
Descripción	El administrador puede desactivar una carta que esté disponible a la venta, y volver a activarla.
Pantalla	Pantalla de Escaparate
Rol	Administrador
Precondición	Estar logeado. La carta no puede ser propiedad de nadie.
Flujo	<ol style="list-style-type: none"> 1) Desde la pantalla de Escaparate, se visualizarán las cartas a la venta. 2) Debajo de cada carta, hay un texto que indica si la carta está disponible o no. 3) Si la carta está desactivada, haciendo clic en el texto, nos pedirá confirmación para activarla y viceversa. 4) Se efectúa la transacción al SC activando o desactivando la carta.
Flujo alternativo	
Postcondición	
Comentarios	

CU7	<i>Compra de una carta</i>
Descripción	Desde el escaparate, el usuario puede comprar una carta y adquirirla en propiedad.
Pantalla	Pantalla de Escaparate
Rol	Usuario y administrador
Precondición	Estar logeado. La carta tiene que estar activada.
Flujo	<ol style="list-style-type: none"> 1) Desde la pantalla de Escaparate, se visualizarán las cartas a la venta. 2) Debajo de cada carta hay un botón de "Comprar" 3) Haciendo clic en el botón de compra, iniciaremos la transacción. 4) Si la transacción ha sido satisfactoria, lanzará un mensaje de "Compra satisfactoria". 5) Si ocurre un problema en la transacción, mostraremos también el mensaje.
Flujo alternativo	
Postcondición	
Comentarios	

CU8	<i>Transferencia de una carta a otro usuario</i>
Descripción	Transferir una carta en propiedad a otro usuario
Pantalla	Pantalla de Mis cartas
Rol	Usuario y administrador
Precondición	Estar logeado. Desde la pantalla de Mis cartas.
Flujo	<ol style="list-style-type: none"> 1) Desde la pantalla de Mis cartas, se visualizarán las cartas del usuario. 2) En la parte inferior de cada carta, representaremos un botón para transferir la carta. 3) Haciendo clic en el botón de transferir, iniciaremos la transacción. 4) Si la transacción ha sido satisfactoria, lanzará un mensaje de "Transferencia satisfactoria" y la carta desaparecerá de la pantalla. 5) Si ocurre un problema en la transacción, mostraremos también el mensaje.
Flujo alternativo	
Postcondición	
Comentarios	

CU9	<i>Poner y quitar la carta a la venta</i>
Descripción	Establecer un precio y poner la carta a la venta para que otro usuario la pueda comprar.
Pantalla	Pantalla de Mis cartas
Rol	Usuario y administrador
Precondición	Estar logeado. Desde la pantalla de Mis cartas.
Flujo	<ol style="list-style-type: none"> 1) Desde la pantalla de Mis cartas, se visualizarán las cartas del usuario. 2) En la parte inferior de cada carta, representaremos un botón para poner la carta a la venta. 3) Haciendo clic en el botón, aparece un formulario para introducir el nuevo precio y un botón de Poner carta a la venta. 4) Haciendo clic en el botón de transferir, iniciaremos la transacción. 5) Si la transacción ha sido satisfactoria, lanzará un mensaje de "La carta está a la venta" y se mostrará el nuevo precio en la carta. 6) Si ocurre un problema en la transacción, mostraremos también el mensaje. 7) Si la carta ya está a la venta, mostraremos la opción de quitar carta de la venta.
Flujo alternativo	
Postcondición	
Comentarios	

CU10	<i>Logout</i>
Descripción	El usuario podrá hacer logout i salir a la pantalla de login.
Pantalla	Cualquier pantalla. Accesible desde opciones del menú principal.
Rol	Usuario y administrador
Precondición	Estar logeado.
Flujo	<ol style="list-style-type: none"> 1) <i>Desde cualquier pantalla.</i> 2) <i>Desplegar el menú principal y haciendo clic en el botón Logout</i> 3) <i>Acceso a la pantalla de login</i>
Flujo alternativo	
Postcondición	
Comentarios	

6 ENTIDADES DEL SISTEMA

Descripción entidades del sistema

1	SC CCGMarketplace
Descripción	CCGMarketplace es el SC principal. Hereda de ERC721. Y este último hereda de IERC721 y de ERC165.
Atributos	<p>Define las propiedades de una carta:</p> <p>struct Token</p> <p>Dirección del propietario:</p> <p>address payable public owner;</p> <p>Mapping para comprobar de forma eficiente si una dirección está asignada a un administrador:</p> <p>mapping(address => bool) public adminByAddress;</p> <p>Balance del contrato:</p> <p>uint public balance;</p> <p>Lista de tokens o cartas:</p> <p>Token[] tokens;</p> <p>Relación de token con su propietario:</p> <p>mapping(uint256 => address) public tokenIdOwnershipOf;</p> <p>Número de tokens de un usuario:</p> <p>mapping (address => uint256) public ownershipTokenCount;</p> <p>Lista de tokens (ids) en venta:</p> <p>uint256[] public tokenIdOnSale;</p> <p>Lista de tokens (ids) por dirección de usuario:</p> <p>mapping(address => uint256[]) public tokenIdByAddress;</p>

2	Usuario / Propietario / Administradores
Descripción	Usuarios (propietarios de cartas) / Propietario del contrato / Administradores del contrato
Atributos	<ul style="list-style-type: none"> • Address

3	<i>Carta</i>
Descripción	<i>NFTs</i>
Atributos	<ul style="list-style-type: none">• <i>Nombre</i>• <i>Descripción</i>• <i>Color</i>• <i>Tipo</i>• <i>Maná</i>• <i>Ataque</i>• <i>Defensa</i>• <i>Precio</i>• <i>Imagen (hash)</i>

6.1 Smart contracts

Nombre SC: <i>CCGMarketplace</i>	
Descripción	Es el contrato principal y contiene todos los métodos y declaraciones públicas.
Relación con otros SC	Hedera de ERC721
Observaciones	

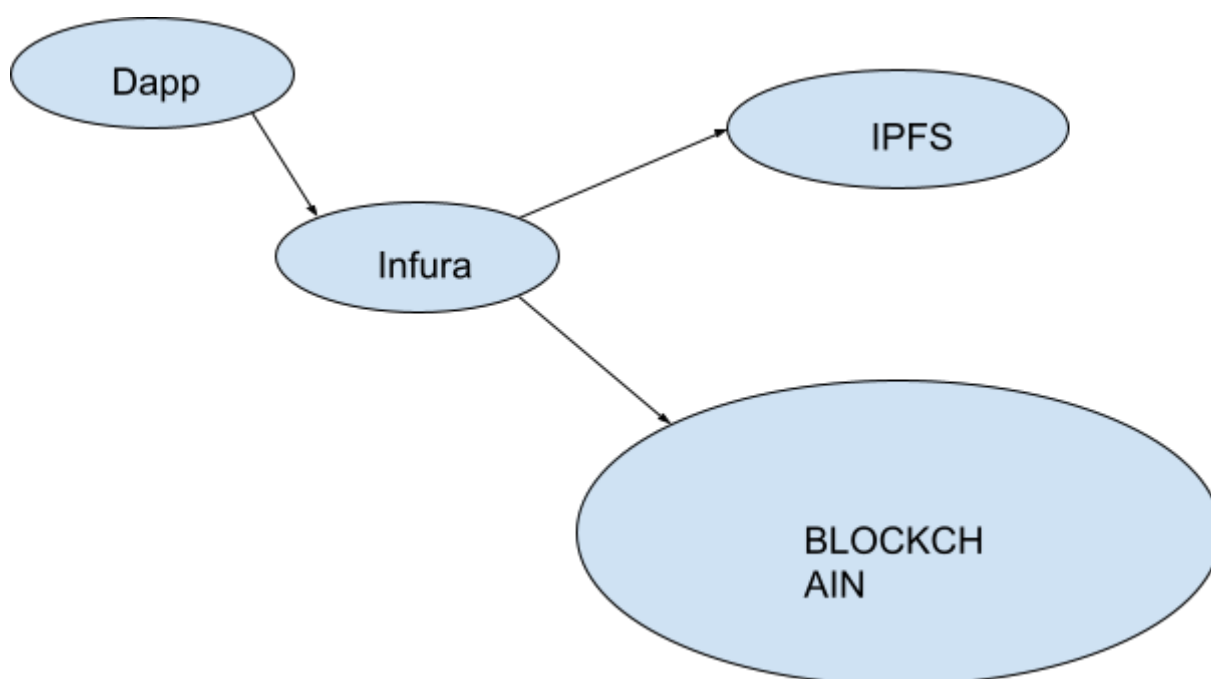
Nombre SC: <i>ERC721</i>	
Descripción	Contiene todos los métodos necesarios para cumplir con la especificación del estándar ERC721 (Non fungible tokens). Contiene todos los métodos definidos en la interfaz IERC721.
Relación con otros SC	Hedera de IERC721 y ERC165
Observaciones	Todos los métodos están declarados y sin implementar. La intención es solamente cumplir con el estándar.

Nombre SC: <i>IERC721</i>	
Descripción	Interface que define los métodos necesarios y obligatorios que hay que implementar para cumplir con el estándar ERC721.
Relación con otros SC	
Observaciones	

Nombre SC: <i>ERC165</i>	
Descripción	El estándar ERC-165 proporciona un estándar para generar un un identificador para una interfaz, en este caso, la interfaz IERC721.. De esta manera, si el identificador de la versión la interfaz IERC721 coincide con su contrato, es una forma de garantizar que su contrato cumpla con esa interfaz. Es una forma de tomar una huella digital de la interfaz y verificarla con una huella digital de las funciones que ha implementado, y asegurarse de que sean las mismas.
Relación con otros SC	<i>IERC721</i>
Observaciones	

Nombre SC: <i>Address</i>	
Descripción	Se utiliza como helper. Contiene métodos relacionados con las direcciones. Para comprobar si la dirección es la de un contrato o para convertir address a address payable, etc.
Relación con otros SC	<i>ERC721</i>
Observaciones	

7 ARQUITECTURA DE LA SOLUCIÓN



8 INFORMACIÓN DE DISPOSITIVOS DE INTEGRACIÓN

Aplicación mobile y web tendrán las mismas pantallas y mismas funcionalidades