

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Sistema de fumigación con drones mediante tecnología blockchain

Javier Martínez Arellano

Experto Universitario en DESARROLLO DE APLICACIONES BLOCKCHAIN

Raúl Jaime Maestre

4 Julio 2022

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Índice de contenido

INTRODUCCIÓN	3
DESCRIPCIÓN DEL CASO	3
ALCANCE SISTEMA DE FUMIGACIÓN	6
<i>Roles</i>	6
<i>Definiciones</i>	6
<i>Consideraciones</i>	7
<i>Historias de usuario</i>	8
JUSTIFICACIÓN DEL USO DE LA TECNOLOGÍA BLOCKCHAIN PARA RESOLVER EL PROBLEMA PROPUESTO	10
<i>Ventajas</i>	10
<i>Desventajas</i>	11
ANÁLISIS Y MODELO DEL SISTEMA DE DESARROLLO UTILIZADO	12
<i>Diagrama de despliegue</i>	12
<i>Diagramas de secuencia</i>	13
<i>Diagrama de secuencia - historia de usuario 1 – Registrar Dron</i>	13
<i>Diagrama de secuencia - historia de usuario 2 – Registrar Parcela</i>	13
<i>Diagrama de secuencia - historia de usuario 3 – Reservar fumigación</i>	14
<i>Diagrama de secuencia - historia de usuario 4 – Fumigar</i>	14
<i>Diagrama de secuencia - historia de usuario 5 – Comprar token FDT</i>	15
<i>Diagrama de secuencia - historia de usuario 6 – Consultar</i>	15
INSTRUCCIONES DE DESPLIEGUE SOBRE LA RED ALASTRIA	16
DESCRIPCIÓN DEL ENTORNO DE DESARROLLO UTILIZADO	24
MANUAL DE USUARIO	28
<i>Historia de Usuario 1 – Proveedor: registro dron</i>	28
<i>Historia de Usuario 2 – Cliente: registro parcela</i>	30
<i>Historia de Usuario 3 – Cliente: reserva fumigación y pagar con tokens FDT(ERC20)</i>	32
<i>Historia de Usuario 4 – Proveedor: consulta y ejecuta fumigaciones pendientes</i>	34
<i>Historia de Usuario 5 – Cliente: usar Exchange para obtener token FDT (ERC20)</i>	36
<i>Historia de Usuario 6 – Cliente: consultar saldo FDT</i>	39
<i>Herramientas de ayuda</i>	40
TESTING	44
CONCLUSIONES	48
ESTRUCTURA DEL PROYECTO	50
ÍNDICE DE FIGURAS	53
ÍNDICE DE TABLAS	55

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Trabajo Final de Experto

Introducción

Este proyecto es un trabajo final del curso de Experto en desarrollo de aplicaciones blockchain. En él, se ha realizado una Dapp (aplicación descentralizada) para la empresa ficticia “Fumigaciones S.A.” que presta sus servicios de fumigación de parcelas por medio de drones.

Para la realización de esta Dapp se ha usado la tecnología blockchain, seguido las metodologías de desarrollo software y testing vistas en el curso así como los principios de buenas prácticas de desarrollo de soluciones blockchain.

Todo el material usado para presentar este proyecto (memoria, documentación, código, imágenes, etc) están disponibles en:

- Google drive (al completo):

https://drive.google.com/drive/folders/1T77P9wbJy-3dTh5n2_fwSyhtbmylnVT?usp=sharing

- Github, sin “node_modules” debido a su gran tamaño:

<https://github.com/jmartinezare/fumdapp>

Descripción del caso

A continuación, se incluye la descripción oficial del proyecto, facilitada por UNIR:

Una empresa ha desarrollado un sistema de fumigación con drones y nos ha solicitado que desarrollemos una solución basada en la blockchain de Alastria para su uso.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Las características propias de los drones son:

- ▶ Un identificador único y ascendente, comenzando en 1 y que no puede repetirse.
- ▶ La empresa que lo gestiona, que será la única que pueda mandar acciones al dron.
- ▶ Altura máxima y mínima de vuelo.
- ▶ Una lista de pesticidas que puede suministrar. Los pesticidas existentes son cinco y sus nombres son: Pesticida A, Pesticida B, Pesticida C, Pesticida D y Pesticida E.
- ▶ Coste.

La operación de fumigación es inmediata y debe lanzar un evento de parcela fumigada con el ID de la parcela.

Las características de las parcelas son:

- ▶ Un identificador único y ascendente, que comienza en 1 y que no puede repetirse.
- ▶ Un propietario.
- ▶ Altura máxima y mínima de vuelo permitida.
- ▶ Pesticida aceptado, que va a ser uno de la lista de pesticidas descrita anteriormente.

Otras operaciones que debe suministrar la plataforma son:

- ▶ Contratar un dron a la empresa para desinfectar una parcela con un pesticida determinado.
- ▶ Pago de la operación realizada desde la cuenta del propietario a la de la empresa.

Para la gestión de pagos se debe crear un token propio basado en el estándar ERC 20. Además, los drones y las parcelas pueden gestionarse mediante tokens no fungibles basados en el estándar ERC 721.

La empresa solicita tener una interfaz web que le permita registrar los drones y asignarles trabajos. A su vez, también se debe proporcionar una interfaz web para

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

que los propietarios de las parcelas las puedan registrar y tengan la posibilidad de contratar un dron con las características que requiere su parcela y que pueda desplazarse hasta la misma.

Cualquier duda que pueda surgir en el desarrollo de este trabajo debe ser consultada con el cliente; en nuestro caso, lo haremos con el director que tengamos asignado.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Alcance Sistema de fumigación

El alcance de este proyecto toma como punto de partida la descripción del caso, descrito en el apartado anterior, y viene determinado por los siguientes apartados:

- ▶ Roles
- ▶ Definiciones
- ▶ Consideraciones
- ▶ Historias de usuarios

Roles

En la siguiente tabla se recogen los roles utilizados en el proyecto:

Núm.	Rol	Descripción
1	Proveedor	Empresa proveedora de servicios de fumigación mediante drones
2	Cliente	Entidad que requiere los servicios del proveedor para fumigar sus parcelas

Tabla 1. Roles

Definiciones

Para facilitar la comprensión del documento, se incluye una serie de definiciones:

- Compatibilidad de un dron y una parcela: un dron es compatible con una parcela cuando su margen de vuelo, definido por el rango de [altura máxima, altura mínima], abarca una parte del rango de vuelo permitido en la parcela, definido por su rango de [altura máxima, altura mínima]. Es decir, puede volar a una determinada altura permitida en la parcela. Además, en la lista de pesticidas disponibles del dron debe contener el pesticida definido para la parcela.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

- Reservar una parcela: implica que se reserva un dron para fumigar una parcela. Dicho dron no podrá ser reservado de nuevo hasta que realice la fumigación. Para poder realizar la reserva, es necesario hacer un pago mediante el token de la plataforma FDT.
- Token FDT: Token ERC20 cuyo símbolo es FDT y su nombre es “Fumigate Dron Token”. Se obtiene en el apartado de Exchange de la plataforma y para ello es necesario realizar un pago en ethers a relación de 1 FDT= 0.001 ETH. Su funcionalidad será la de permitir hacer reservas. Los drones tiene un coste de fumigación indicado en este token.
- Usuario de la interfaz web: puede ser tanto el proveedor como cualquiera de los clientes. El proveedor también puede actuar como cliente si así lo desea.
- Dapp: aplicación descentralizada
- Proveedor/dueño: En este proyecto, sinónimos que hacen referencia el propietario de la Dapp, la empresa que provee el servicio de fumigación con drones.

Consideraciones

A continuación, se detalla una lista de consideraciones que se han tenido en cuenta en el proyecto para su implementación:

- La contratación de un dron queda abierta a cualquier usuario (cliente o proveedor) ya que se entiende que no hay ninguna restricción sobre ello en el enunciado.
- El dueño tiene la capacidad de actuar también como cliente, es decir, puede utilizar las mismas acciones reservadas para clientes.
- Un cliente, solo puede reservar fumigaciones para sus parcelas.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

- Un cliente, no puede ejecutar una fumigación reservada (mostrado en el manual de usuario), solo el proveedor puede hacerlo.
- Si un cliente no tiene saldo del token FDT, no podrá reservar un dron para fumigar una de sus parcelas (mostrado en el manual de usuario).
- Al comprar tokens FDTs, es necesario realizar un pago, a la cuenta del contrato, en ETH con la equivalencia 1 FDT = 0.001 ETH.
- Si el cliente no tiene saldo suficiente de ETH para comprar el número de tokens FDTs seleccionado, no podrá hacerlo.
- Al reservar una fumigación, se debe realizar un pago en FDT, de la cuenta del cliente a la del contrato, equivalente al coste definido en el dron reservado.
- Al comprar FDT, se produce un envío de tokens de la cuenta del contrato a la del cliente con las unidades de token FDT compradas, produciéndose un incremento en el balance de la cuenta del cliente y una disminución en la cuenta del contrato (mostrado en el manual de usuario)

Historias de usuario

A continuación, se describen las historias de usuario más importantes y que definen el alcance de este proyecto:

Núm.	Definición	Descripción
1	Proveedor: registro dron	<p>Proveedor registra un dron con los siguientes datos:</p> <ul style="list-style-type: none"> • Altura máxima • Altura mínima • Pesticidas disponibles (permite multi-selección) • Coste de fumigación medido en tokens FDT (ECR20)

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

		<p>Al registrar un dron, se da de alta como un token ERC721</p> <p>El identificador es automático, se auto-incrementa (no se repite) y comienza en 1</p>
2	Cliente: registro parcela	<p>Cliente registra una parcela con los siguientes datos:</p> <ul style="list-style-type: none"> • Altura máxima • Altura mínima • Pesticida aceptado <p>Al registrar una parcela, se da de alta como un token ERC721</p> <p>El identificador es automático, auto-incremental (no se repite) y comienza en 1</p>
3	<p>Cliente: reserva fumigación y paga con tokens FDT(ERC20)</p>	<ol style="list-style-type: none"> 1- Cliente consulta parcelas disponibles de su propiedad 2- Busca automática de un dron disponible y que sea compatible con su parcela 3- Reserva la fumigación de la parcela con el dron obtenido <p>La reserva se hace por medio del token propio FDT (Fumigate Dron Token) ERC20 que se adquiere en la misma web</p>
4	Proveedor: consulta y ejecuta fumigaciones pendientes	Proveedor consulta todas las fumigaciones disponibles, elige una y la ejecuta. Conlleva que se liberen la parcela y el dron para futuras reservas.
5	Cliente: usa Exchange para obtener token FDT (ERC20)	Cliente obtiene tokens FDT, Fumigate Dron Token, (ERC20) que permiten la reserva de fumigaciones para una parcela.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

		Para obtenerlos es necesario su pago en ethers a razón de 1 FDT = 0.001 ETH más el gas correspondiente a la transacción
6	Cliente: consultar saldo FDT	Cliente consulta saldo de tokens FDT disponible en una cuenta

Tabla 2. Historias de usuario

Justificación del uso de la tecnología blockchain para resolver el problema propuesto

Blockchain es un libro mayor distribuido, con información de transacciones, creado por bloques ordenados cronológicamente que forman una cadena, de ahí su nombre. Este proyecto abarca una funcionalidad muy reducida y solo de la parte basada en blockchain pero se estima que, para un proyecto real deberían usarse soluciones mixtas que terminasen de agregar ese valor añadido con datos sensibles, fuera de la cadena, necesarios para poder cumplir las leyes fiscales vigentes.

En relación al alcance de este proyecto se definen algunas ventajas y desventajas que tiene el afrontar este proyecto con tecnología blockchain:

Ventajas

- Persistencia infinita. Mientras haya nodos que formen parte de la red, permanecerá la información en la red.
- Transparencia: cada nodo en la red tiene una copia de la información a la que se puede consultar y auditar en tiempo real.
- Inmutabilidad. Uno de los pilares y principios de la tecnología blockchain es no poder modificar un bloque que ya haya sido minado, es decir, incorporado a la red. Esto añade un gran valor de seguridad y de confianza a los usuarios ya que un dato añadido a la cadena es un dato que permanece inalterable.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

- Integridad de los procesos: los protocolos de las principales redes son abiertos y se puede comprobar el funcionamiento real de los mismos.
- Copias de seguridad implícitas: por lo descrito anteriormente, existe una redundancia de datos suficiente como para tener la garantía de que los datos están seguros en la red.
- Este sistema aporta una gran sencillez de uso ya que solo es necesario tener cuentas de Eth y FDT para poder ofrecer y utilizar este servicio.
- Inmediatez. La reserva de una fumigación es inmediata, sin burocracia, colas de espera, etc. Solo es necesario que haya drones disponibles
- Igualdad: Cualquier personal que reúna las condiciones del punto anterior puede utilizar el servicio en igual de condiciones que otro usuario
- Globalización. No hay restricciones geográficas o culturales. Si hay disponible un dron (supuestamente en tu zona), tienes servicio.

Desventajas

- Para que una red blockchain pueda existir es necesario una infraestructura tecnológica importante con un número mínimo de nodos que den soporte a la misma. Esto puede ser un requisito bastante difícil de salvar para entornos con pocos recursos pero podría ser solucionado utilizando una red ya existente pero que también sea utilizada con otros propósitos distintos a los del proyecto.
- La cadena de bloques, al ser de todos y de nadie, supone un gran reto legal en la definición de obligaciones y responsabilidades que conlleva la red y los datos que contiene. Sería muy recomendable que existiesen contratos tradicionales que ayudasen a definir las condiciones del servicio, los derechos y las obligaciones de las partes.
- La inmutabilidad de la red puede ser incompatible con otros derechos fundamentales de los usuarios, como puede ser el del derecho al olvido o privacidad, si lo publicado por ellos mismos o por otros infringiese su

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

intimidad. Esto podría producirse si se almacenasen datos sensibles de parcelas, drones, clientes, etc.

Para propósitos académicos este proyecto ayuda a comprender las ventajas y limitaciones de los desarrollos realizados utilizando la tecnología blockchain.

Análisis y modelo del sistema de desarrollo utilizado

Diagrama de despliegue

En este apartado se incluye el diagrama de despliegue del proyecto. Los contratos se han agrupado en dos partes: los que son genéricos, es decir, que ya existían en otras librerías públicas como Openzeppelin y, por otro lado, los desarrollados para este proyecto.

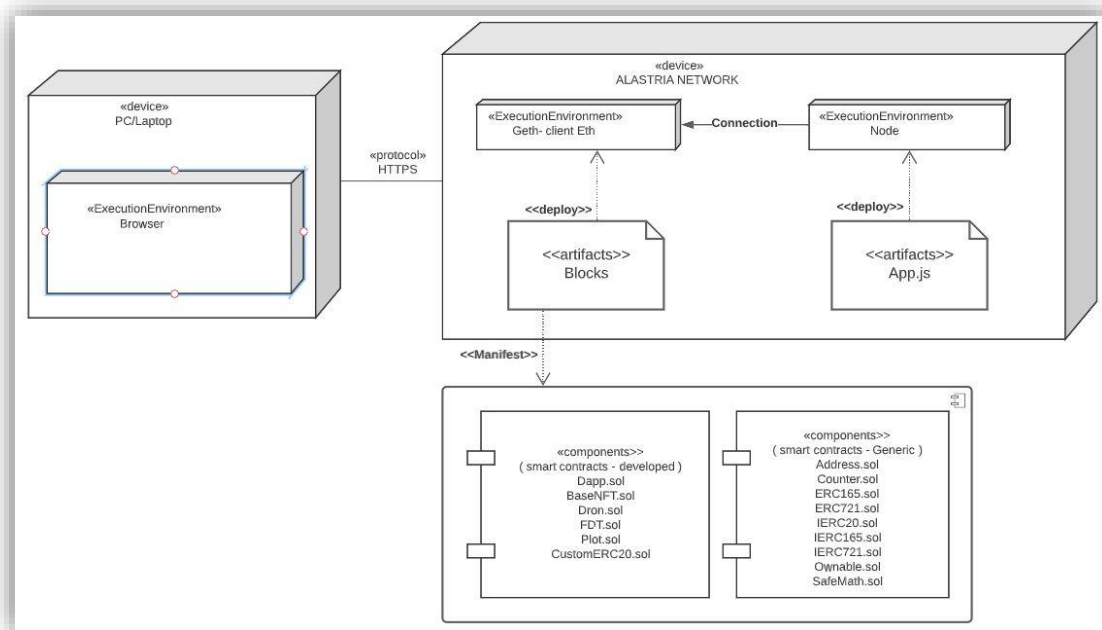


Figura 1. Diagrama de despliegue

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Diagramas de secuencia

En los siguientes apartados se irán reflejando los diagramas de secuencia correspondientes a las historias de usuario definidas en el actual proyecto.

Diagrama de secuencia - historia de usuario 1 – Registrar Dron

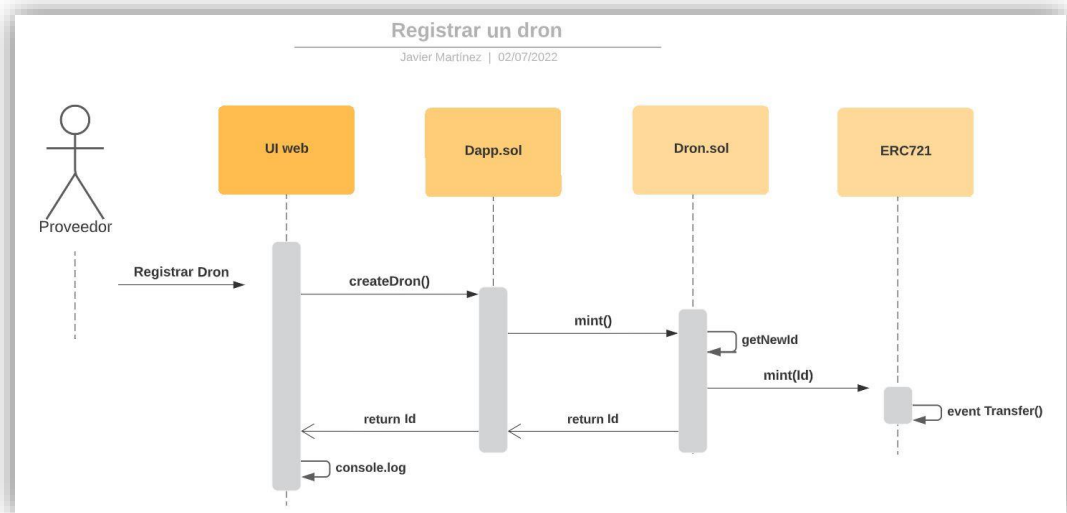


Figura 2. Diagrama secuencia - Registrar dron

El evento de ERC71 sería el movimiento, a la cuenta del propietario, del token creado.

Diagrama de secuencia - historia de usuario 2 – Registrar Parcela

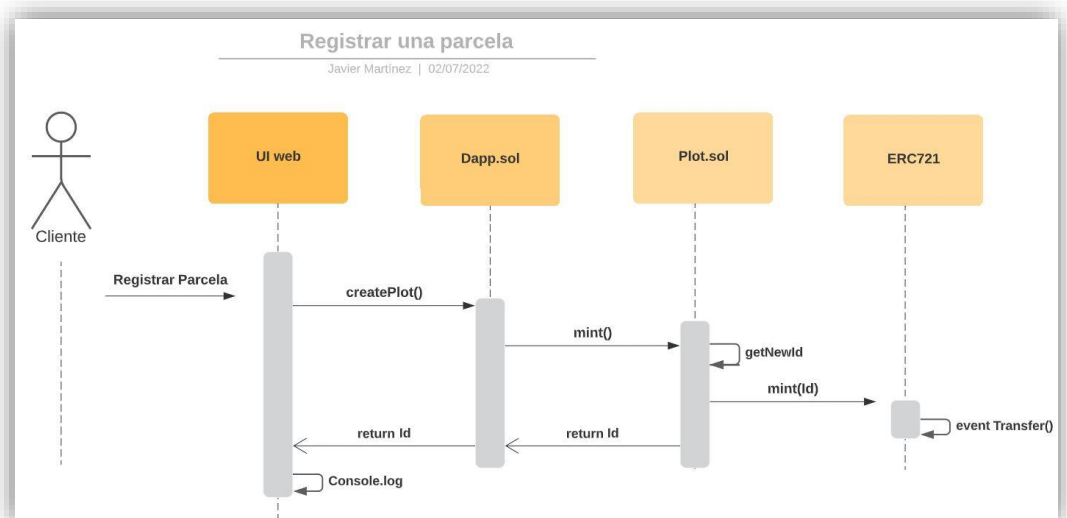


Figura 3. Diagrama secuencia - Registrar parcela

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Diagrama de secuencia - historia de usuario 3 – Reservar fumigación

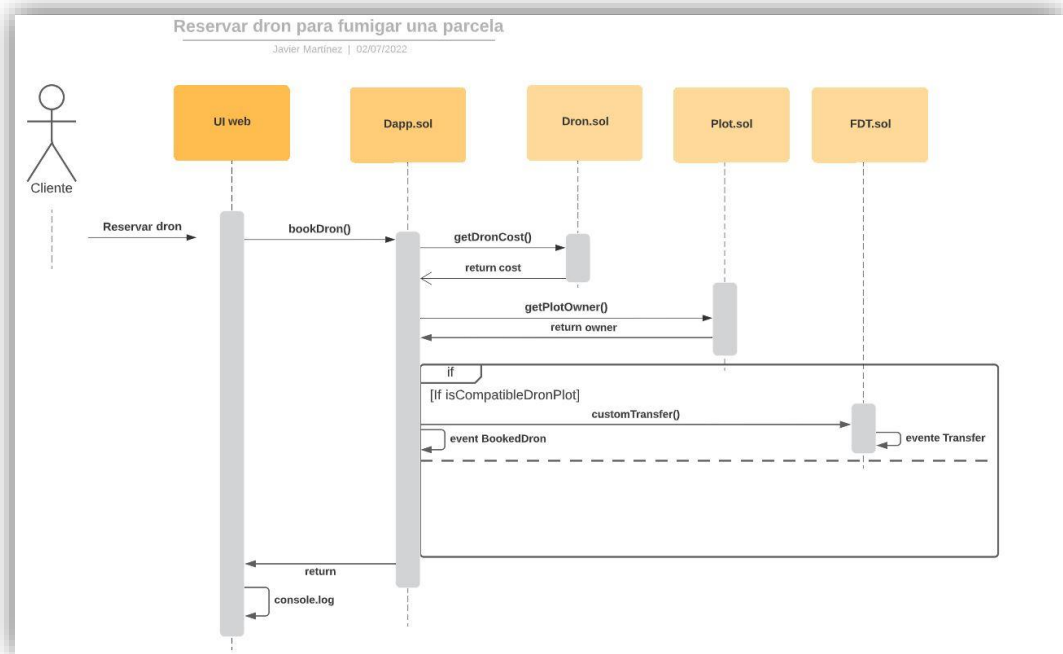


Figura 4. Diagrama secuencia – Reservar fumigación

Diagrama de secuencia - historia de usuario 4 – Fumigar

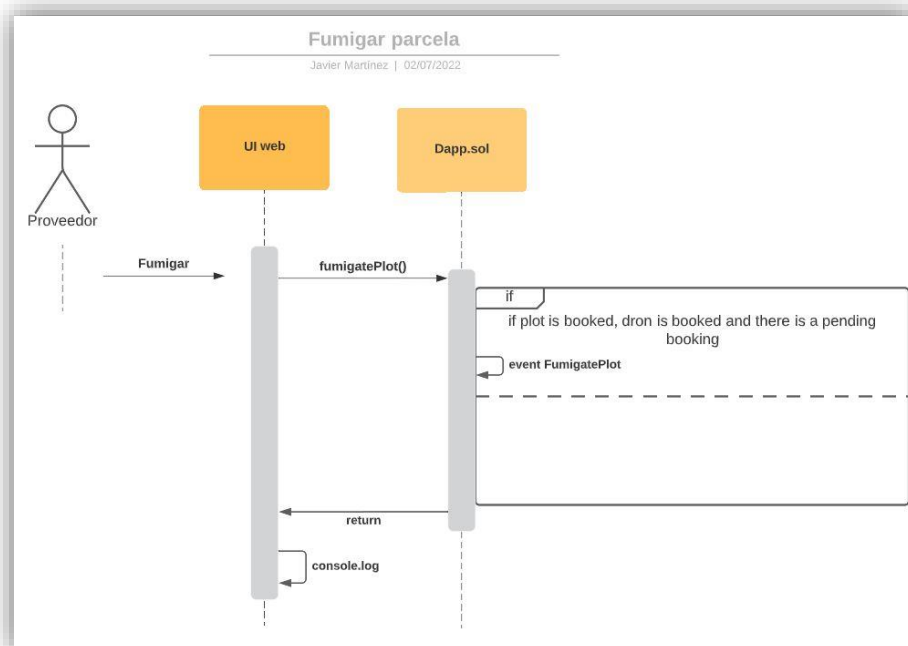


Figura 5. Diagrama secuencia - Fumigar

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Diagrama de secuencia - historia de usuario 5 – Comprar token FDT

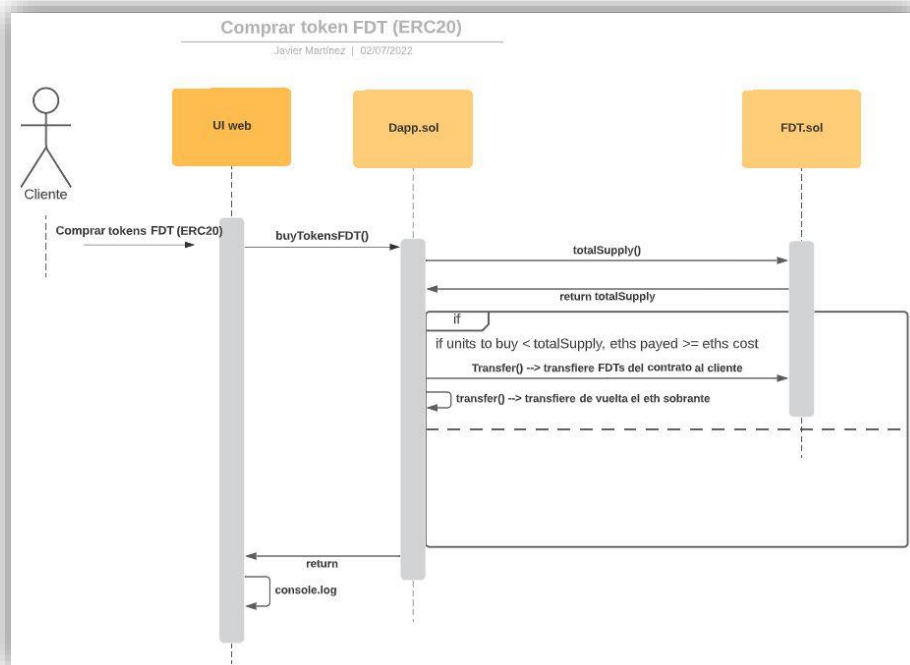


Figura 6. Diagrama secuencia – Comprar token

Diagrama de secuencia - historia de usuario 6 – Consultar

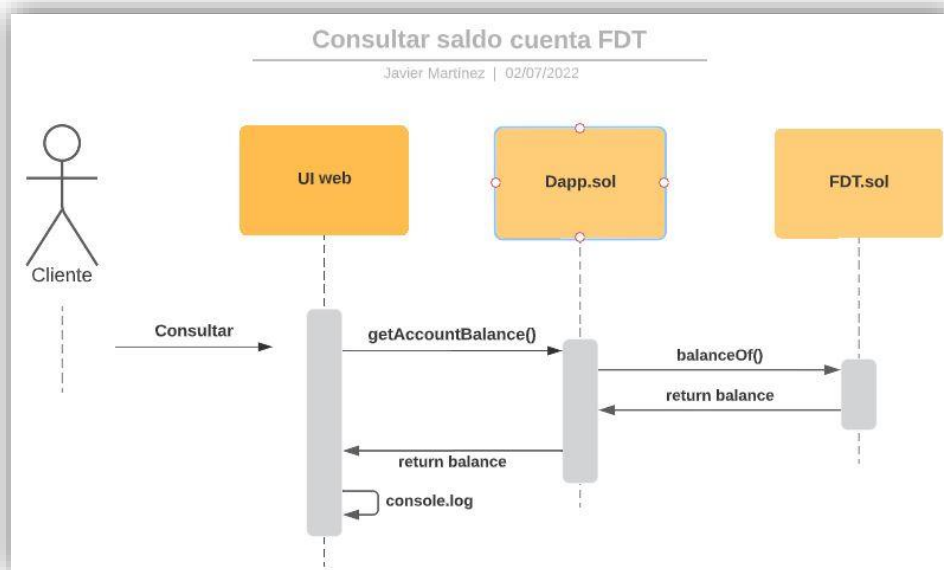


Figura 7. Diagrama secuencia – Consultar saldo

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Instrucciones de despliegue sobre la red Alastria

A continuación, se indica cómo realizar el despliegue de la solución en la red de Alastria por medio de Remix.

Es necesario realizar los siguientes pasos en un dispositivo con acceso a esta red. Para conectarse a esta red se ha utilizado el VDI proporcionado por UNIR con una máquina virtual “AlastriaTelsius – Oracle VM Virtualbox” que tiene acceso a dicha red.

Los elementos utilizados para este despliegue son:

- Sistema operativo Ubuntu 16.04 LTS
- Navegador utilizado: Firefox 76.0.1 (de 64 bits)
- Terminal de comandos

Primer paso es conectarse, por medio de un cliente, a la red. En este caso usaremos el cliente geth. Para ello, se ejecuta el siguiente comando:

```
$ sudo geth attach http://10.141.8.11:8545
```

A continuación, se pueden consultar las cuentas disponibles:

```
$ web3.eth.accounts
```


Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

```
> web3.eth.accounts
["0x00e553a9c7c283f024a1ae074a191246460a1f40", "0x2db7002474e1b361cf2fc57c9a19c0aa936fe01b", "0xbc869c21d631e122d35789942a573241ec04d2e4", "0xad11f232919a54696791387e3a74a63394c2dafb", "0x35ad6e72cb2ec714b80154b796c7835f97053d3e", "0xa3fef7d78a13f6b6bb1cf60c20bb854c7ed2d8d17", "0x09702705ebcd2c925b3c56662e4982ebec8bce7d", "0xc35fdb9f41a34e998f4d094922e190b4c6fd8e32", "0x11c5395d602289b7407ceebb4fdd5707772c6ae", "0x48d095879b4ebde16b74129c4ec9d3d78d984b80", "0xea66394b0ecc0175b7b4889a24ea959e84b2d32c", "0x241bae338d230276f8bcd38e7ac7bc6a1cbdf22", "0xdd662e2eda6cb3418d4b06541259db3a660b9e97", "0x5d51287ba94ae76bac48776c707547b95bd54db8", "0x1e70df931eba19417791414604c27263098c1b5a", "0x1cc5afcb1b718768bc2f34b6fc85c9bb884f0d937", "0xc2fbc5c89bd4c92a5f36bae92a9c77e3975acdf2", "0xa36ac0371358851548ced85da9930aef0394bf03", "0x7e6a4b8492202e32e168d77b17bc46450f6f8dc4", "0x91a934043c8b15595b1d8a6bc29d04d377332eba", "0xdfe3dbe29b3bba4f567962a929a55712acef2d90", "0xec42a16f289e03746df4fcf1f7c89739b1257f7d", "0x0b0cd7bada2347ccf0db263f30cfe0c43b666dc5", "0xd27e555e82eb8bcc92dc15d5609ad9779710bf81", "0x6eb695b5c6b5c62d6273e0fb049d18604a0afc54", "0x4d0a371ec375130316378c1623eedaf7ea7d9b5f", "0x50c902b417f85b3c1500dbe2c40ae7963de77680", "0x4cfa5f3a9e25db9e61d7b2cfe9086f4625fad810", "0x50d32382ea6304ce41b959c83b1f07b5cc6f731a", "0xe96c52d1d7a489ca7ee1a8145c6f7bf2b247f3fa", "0x5c1a6755ea6d47776fb2738139fac30c0944ed9c", "0x4f8a337e27f7b97b92c15e89fe788f597409e857", "0x088070def5301b950663f6d901f7d48f1326a310", "0xdb687a96a00849cf20eaf5887ef590d2732d55fa", "0x466f33bef44c3084f959c69d9e7ad6b04523a8a0", "0xabef03cd1d3e702412a32f9894dd25be176eaba5"]
```

Figura 8. Cuentas disponibles

Para este proyecto se utilizará la siguiente cuenta, destinada a los alumnos del curso:

"0xbc869c21d631e122d35789942a573241ec04d2e4"

Con el siguiente comando se desbloquee la cuenta:

```
$ web3.personal.unlockAccount
("0xbc869c21d631e122d35789942a573241ec04d2e4", "4mFmfbLsSIUS9b5msSfx", 0);
```

Posteriormente, el paso a seguir es abrir el editor online Remix. La url es

<http://remix.ethereum.org>

Es muy importante utilizar el protocolo http y no https o no se podrá conectar a la red en pasos posteriores.

En el apartado del compilador, se deberá seleccionar la 0.5.1, que fue el usado durante el desarrollo.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

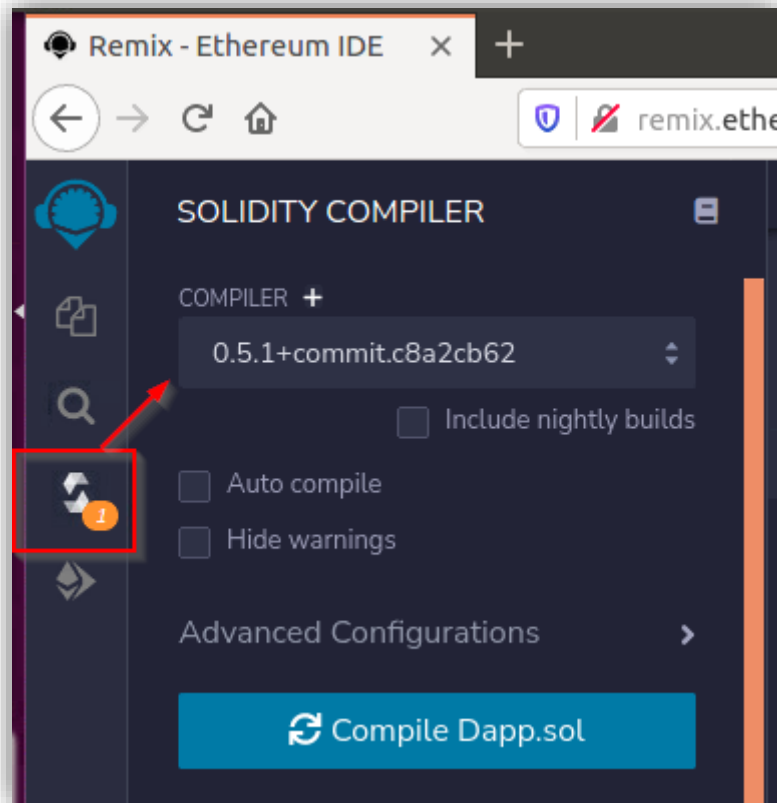


Figura 9. Versión del compilador

A continuación, con el fichero Dapp.sol abierto, se ejecuta el botón de “Compilar Dapp.sol”

Ahora es el momento de realizar el despliegue en la red, para ello es necesario ir al apartado “Deploy & Run Transactions” donde se deberá elegir la cuenta que hemos desbloqueado previamente (terminada en “4D2E4”) y en “Environment” elegir “Web3 Provider”.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

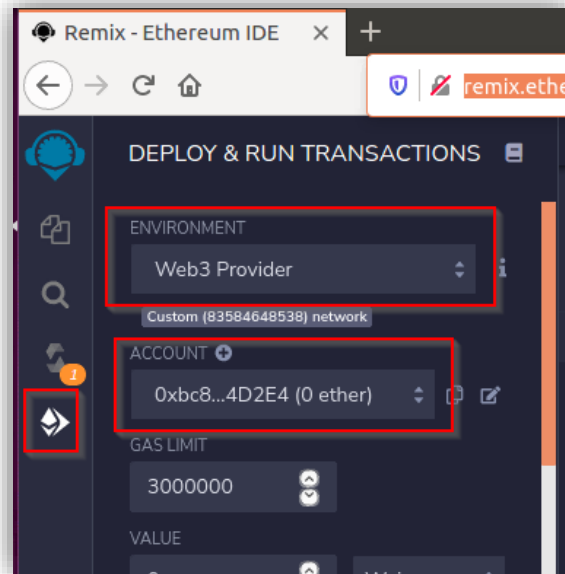


Figura 10. Valores de Environment y Account

Al seleccionar el valor de “Environment”, aparecerá una pantalla como la siguiente donde se debe indicar el siguiente valor para el EndPoint:

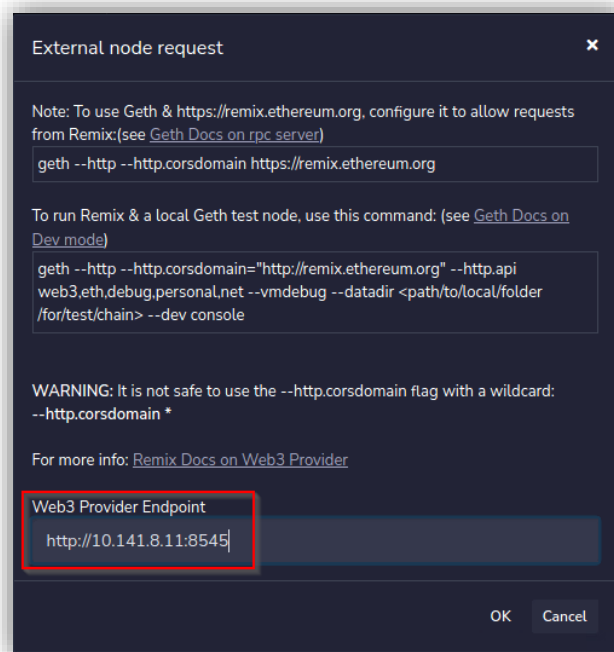


Figura 11. EndPoint Remix, conexión con Alastria

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Una vez hecho esto, aparecerán disponibles en “Account” las cuentas que hemos consultado por consola. Se debe seleccionar la mencionada anteriormente y a continuación se debe pulsar sobre el botón “Deploy”. El resultado debería ser similar al mostrado en la siguiente imagen donde se puede observar, en la parte derecha, el resultado de la transacción correctamente ejecutada:

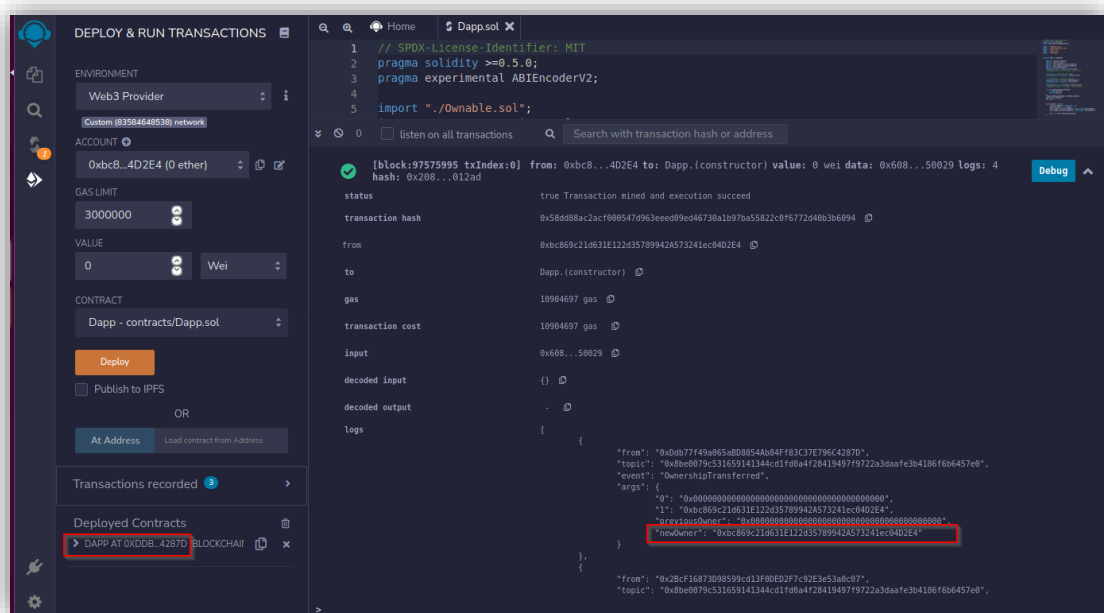


Figura 12. Contrato desplegado

Se puede ver que ha asignado la cuenta que hemos usado como owner del contrato y también, a la izquierda, se indica la cuenta del contrato que se ha creado:

En este caso nos muestra: 0xDdb77f49a065aBD8854Ab84Ff83C37E796C4287D

Ejemplo de uso del contrato desplegado, registrar un dron:

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

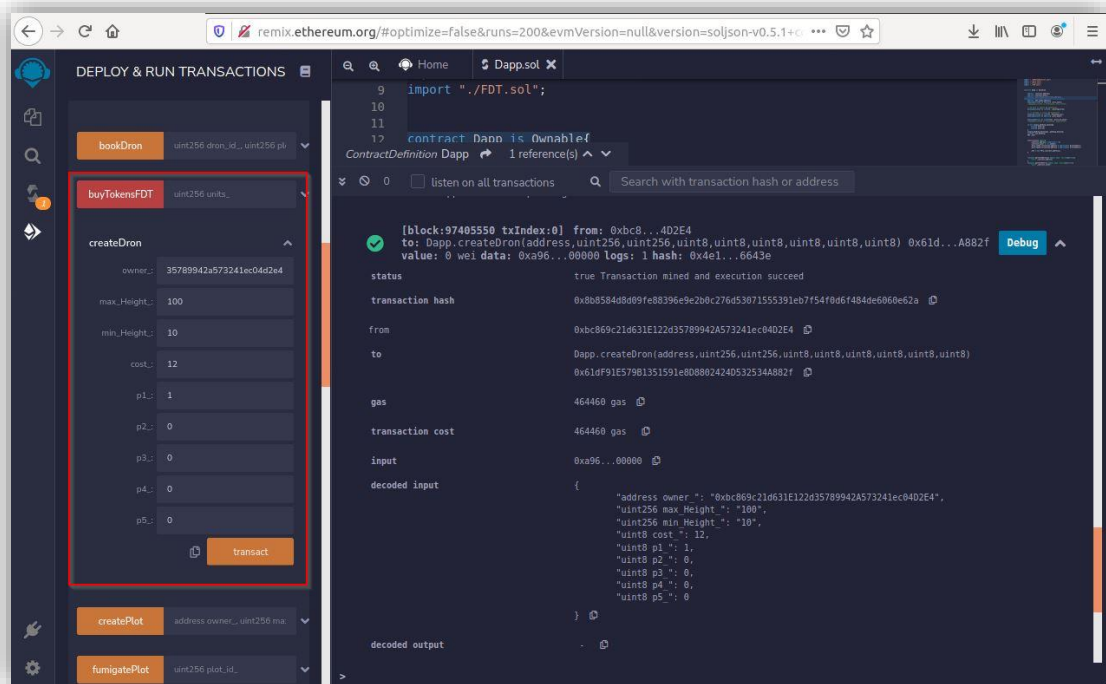


Figura 13. Ejecución de buyTokenFDT

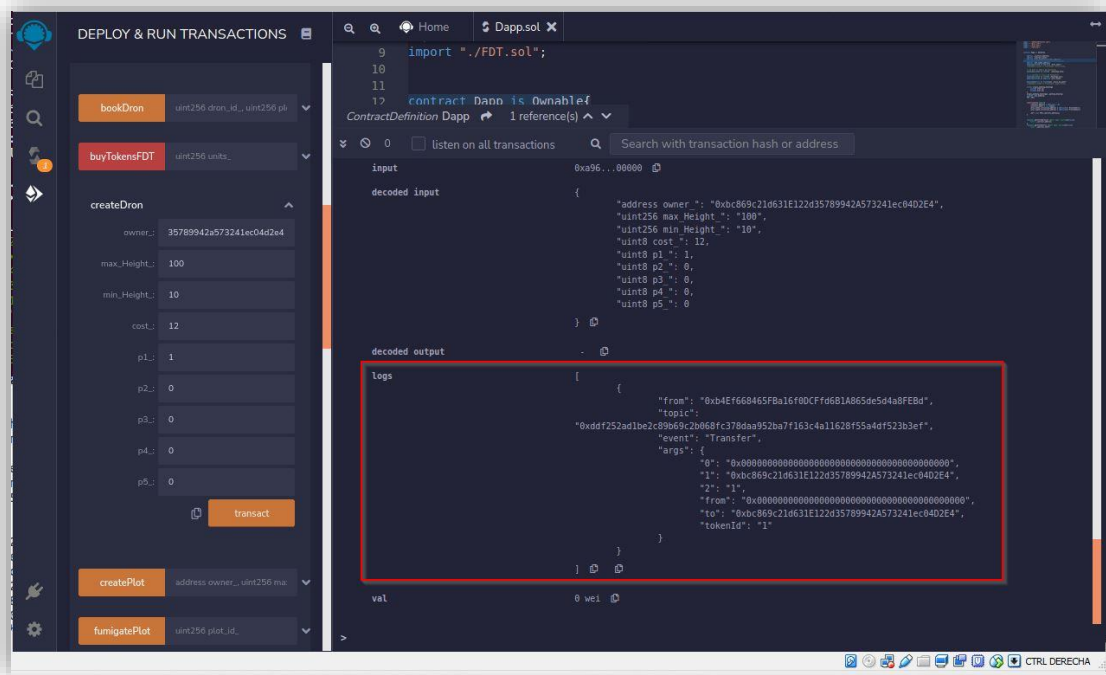


Figura 14. Log ejecución buyToken FDT

Otro ejemplo, obtener información de un dron:

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

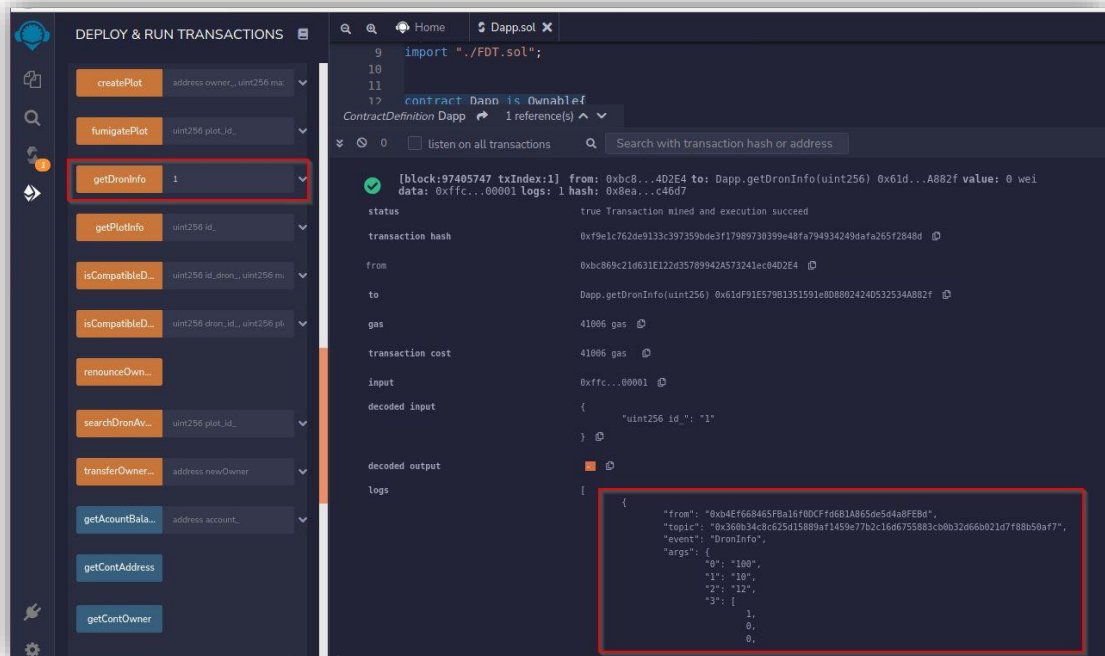


Figura 15. Ejecución getDronInfo

Nota: para el desarrollo se utilizó una funcionalidad que para la versión 0.5 de solidity la muestra como experimental y que actualmente, desde la versión 0.8, ya no es experimental. El motivo buscado era poder devolver en una función un array de valores.

Por este motivo, al compilar muestra el único Warning que se obtiene en este proyecto:

```
contracts/Dapp.sol:3:1:
Warning: Experimental features
are turned on. Do not use
experimental features on live
deployments.
pragma experimental
ABIEncoderV2;
^-----
_ ^
```

Figura 16. Aviso al compilar

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

La solución desarrollada también incluye una interfaz web. Par poder utilizarla será necesario disponer de React. En el siguiente apartado se incluye los pasos necesarios para poder instalar React y poder hacer uso de este interfaz. Su funcionalidad se detalla en el manual de usuario.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Descripción del entorno de desarrollo utilizado

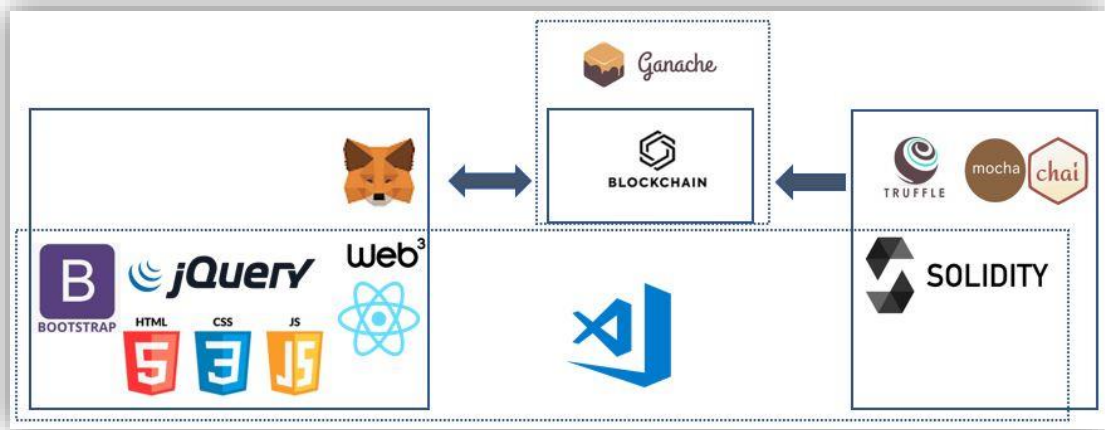


Figura 17. Conjunto de herramientas de desarrollo (Elaboración propia)

En este apartado se mostrará cómo se ha montado y configurado el entorno donde se ha realizado el desarrollo del proyecto, en el que también se incluye levantar un servidor web con React para poder utilizar la interfaz web realizada para este proyecto, más amigable que la proporcionada por Remix.

- Como entorno de desarrollo (IDE) se ha elegido Visual Studio Code 1.68.1, en adelante se mencionará como VSC.
- Extensiones de VSC: solidity v0.0.139 que ayuda con la sintaxis.
- La versión de Solidity (pragma) utilizada parte de la 0.5.0

Se ha elegido esta versión ya que fue usada en prácticas anteriores en la que se hacían despliegues sobre la red de Alastria y se comentó en las clases que era adecuada para este fin. Esto condiciona de forma muy importante la versión de otros paquetes de herramientas, pero se ha decidido priorizar esta versión para que se pueda desplegar en Alastria.

- @openzeppelin/contracts, v2.3.0: Conjunto de contratos que han sido utilizados y testeados por la comunidad. Es muy importante que la versión no

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

sea la más reciente ya que crea incompatibilidades difíciles de detectar. Para instalar esta versión, se puede usar el siguiente comando:

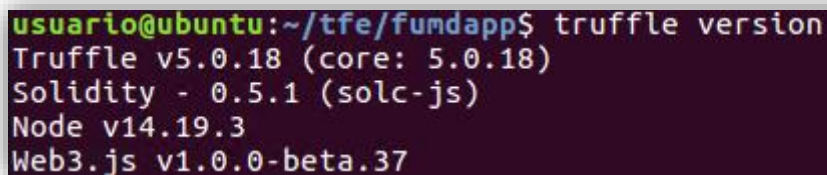
```
$ npm install @openzeppelin/contracts@v2.3.0
```

- Truffle v5.0.18: Suite de herramientas que nos permite, entre otras funcionalidades, compilar contratos, desplegarlos en distintas redes y realizar test.

Como se ha comentado anteriormente, aquí también es muy importante la versión.

```
$ npm install -g truffle
```

```
$ truffle version
```



```
usuario@ubuntu:~/tfe/fumdapp$ truffle version
Truffle v5.0.18 (core: 5.0.18)
Solidity - 0.5.1 (solc-js)
Node v14.19.3
Web3.js v1.0.0-beta.37
```

Figura 18. Versión de truffle

Para crear un proyecto truffle, situados sobre la raíz del proyecto, se haría de la siguiente manera:

```
$ truffle init
```

- React v18.2.0: Nos permitirá realizar una interfaz web, así como levantar un servidor web, con la que interactuar con los contratos realizados.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

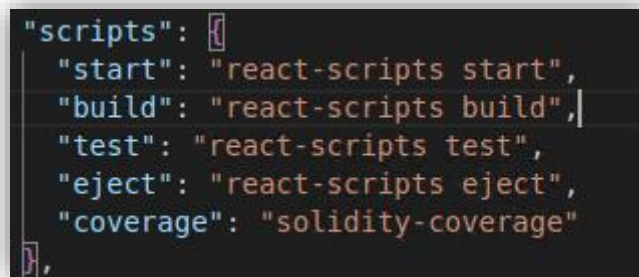
```
$ npm install react
```

```
$ npm install react-scripts@4.0.3
```

Para levantar el servidor web:

```
$ npm run start
```

En la configuración del proyecto, fichero package.json, se deben configurar los comandos necesarios para poder realizar esta acción de start y, test para las pruebas que se muestran en apartados posteriores:



```
"scripts": {
  "start": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject",
  "coverage": "solidity-coverage"
},
```

Figura 19. Valores de Scripts en fichero package.json

- Bootstrap, Popper.js y jquery: necesarios para maquetar y dar funcionalidad a la interfaz web:

```
$ npm install bootstrap jquery popper.js --save
```

- Ganache-cli herramienta necesaria para montar una red blockchain de pruebas sobre la que desplegar los Smart Contracts durante el proceso de desarrollo. Para su instalación, se ejecutaría lo siguiente:

```
$ npm install -g ganache-cli
```

Para crear la red se utiliza el siguiente comando:

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

```
$ ganache-cli --deterministic --networkId 5777 -l 6000000000
```

En este comando, mediante el parámetro “--deterministic” se especifica que ganache genere siempre las mismas cuentas. Con “--networkId 5777”, se define el id de la red y con “-l 6000000000” el límite de gas. El gas será importante para evitar futuros errores de despliegue según el proyecto vaya creciendo.

Con esto, las cuentas que nos genera ganache-cli son las siguientes:

Available Accounts

=====

- (0) 0x90F8bf6A479f320ead074411a4B0e7944Ea8c9C1 (100 ETH) → **Proveedor**
- (1) 0xFFcf8FDEE72ac11b5c542428B35EEF5769C409f0 (100 ETH) → **Cliente**
- (2) 0x22d491Bde2303f2f43325b2108D26f1eAbA1e32b (100 ETH)
- (3) 0xE11BA2b4D45Eaed5996Cd0823791E0C93114882d (100 ETH)
- (4) 0xd03ea8624C8C5987235048901fB614fDcA89b117 (100 ETH)
- (5) 0x95cED938F7991cd0dFcb48F0a06a40FA1aF46EBC (100 ETH)
- (6) 0x3E5e9111Ae8eB78Fe1CC3bb8915d5D461F3Ef9A9 (100 ETH)
- (7) 0x28a8746e75304c0780E011BEd21C72cD78cd535E (100 ETH)
- (8) 0xACa94ef8bD5ffEE41947b4585a84BdA5a3d3DA6E (100 ETH)
- (9) 0x1dF62f291b2E969fB0849d99D9Ce41e2F137006e (100 ETH)

La primera será la usada para el proveedor y la segunda para el cliente

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Manual de usuario

Historia de Usuario 1 – Proveedor: registro dron

Figura 20. Interfaz web: Registro dron

El Proveedor da de alta un dron mediante su registro en la flota de Fumigaciones S.A. Para ello, rellena los datos que definen al dron y ejecuta su transacción. El Id obtenido es automático y no lo elige el usuario. Comienza en 1 y es auto-incremental. Esta acción de registrar un dron conlleva una transacción con su coste de gas correspondiente.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

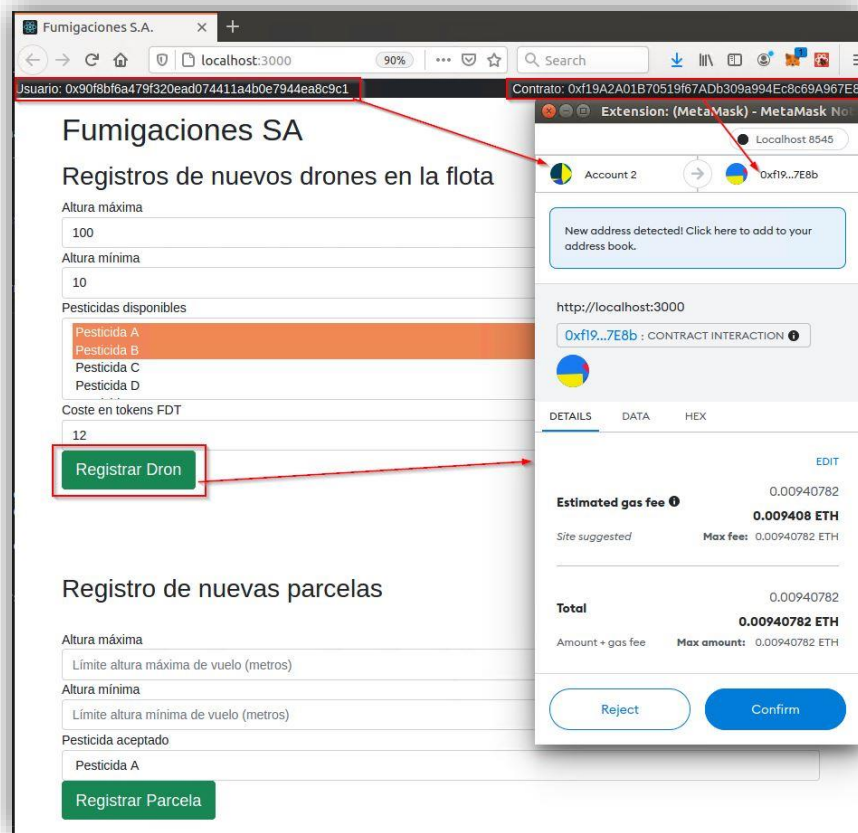


Figura 21. Interfaz web: Registro dron-transacción Metamask

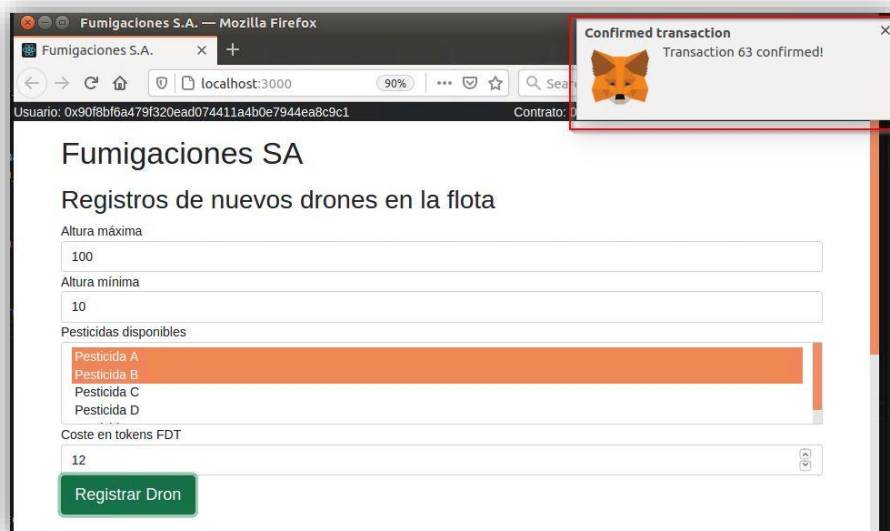


Figura 22. Interfaz web: Registro dron-confirmación transacción

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Historia de Usuario 2 – Cliente: registro parcela

El cliente registra una parcela en la plataforma. Para ello, rellena los datos que definen a la parcela y ejecuta la transacción. El Id obtenido es automático y no lo elige el usuario. Comienza en 1 y es auto-incremental. Esta acción de registrar parcela conlleva una transacción con su coste de gas correspondiente.



Registro de nuevas parcelas

Altura máxima
50

Altura mínima
10

Pesticida aceptado
Pesticida B

Registrar Parcela

Figura 23. Interfaz web: Registro parcela



Registro de nuevas parcelas

Altura máxima
50

Altura mínima
10

Pesticida aceptado

Pesticida B

Pesticida A

Pesticida B

Pesticida C

Pesticida D

Pesticida E

Figura 24. Interfaz web: Registro parcela-valores pesticida aceptado

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

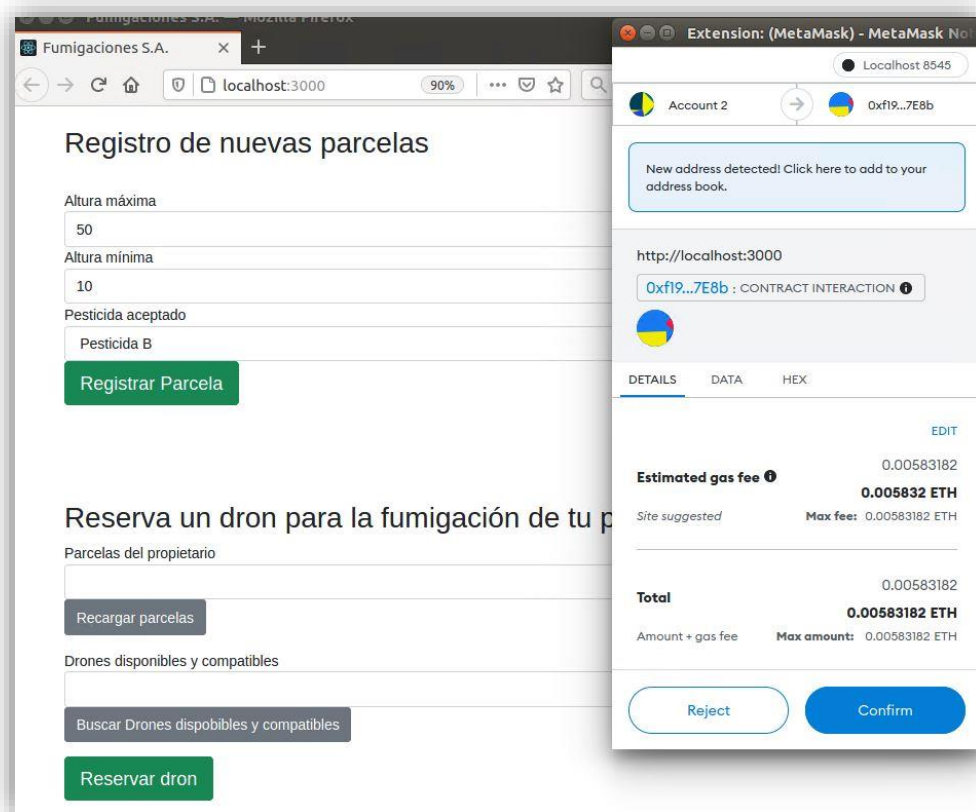


Figura 25. Interfaz web: Registro parcela-transacción Metamask

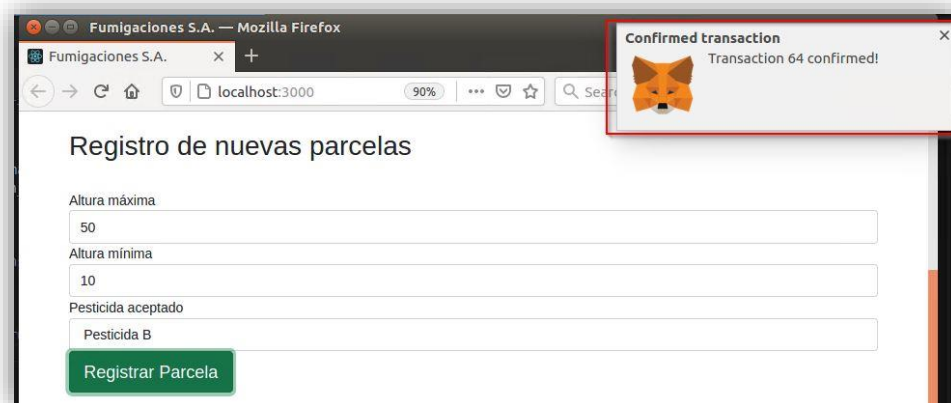


Figura 26. Interfaz web: Registro parcela-confirmación transacción

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Historia de Usuario 3 – Cliente: reserva fumigación y pagar con tokens FDT(ERC20)

El cliente consulta las parcelas de su propiedad, mediante la acción de “Recargar parcelas”. En el desplegable aparecerá una lista con todas sus parcelas. Elige una y ejecuta la acción “Busca Drones disponibles y compatibles”. Desde la interfaz web, en los desplegables se limita los valores que se pueden utilizar. El usuario no pueden introducir en ellos valores manualmente.

Esto busca los drones dados de alta en la plataforma que no estén reservados para fumigar otra parcela y que además sea compatible con la parcela. La definición de compatibilidad de un dron y una parcela se puede encontrar en el apartado de definiciones. Esta acción de reserva conlleva una transacción con un coste en tokens FDT a relación de 1 FDT = 0.001 ETH, más un coste de gas correspondiente.

Figura 27. Interfaz web: Reservar

Figura 28. Interfaz web: Reservar-valores parcela

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

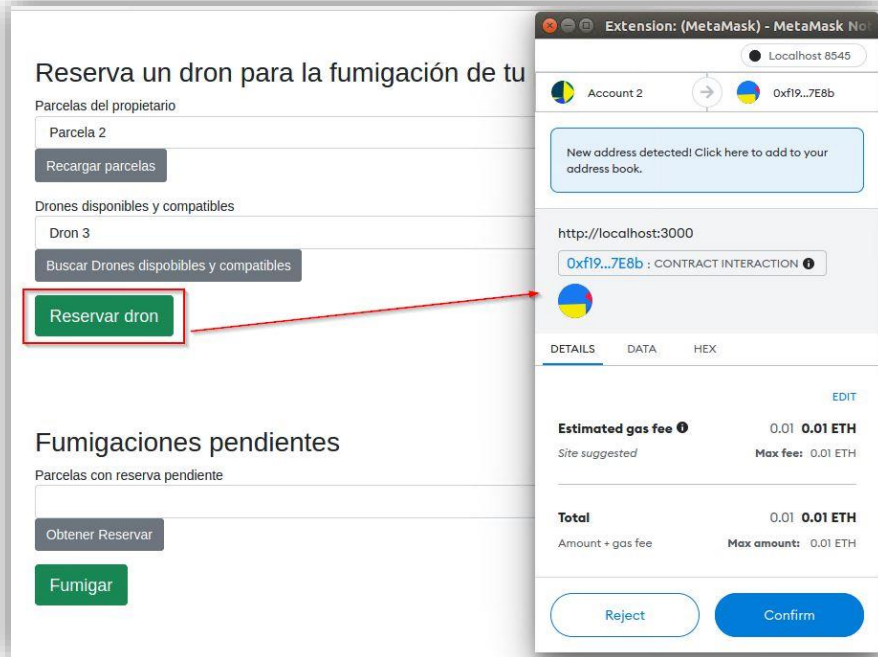


Figura 29. Interfaz web: Reservar – transacción Metamask



Figura 30. Interfaz web: Reservar-confirmación transacción

En caso de que un cliente intente hacer una reserva y su saldo en tokens FDT sea inferior al coste indicado en el dron, el sistema revertirá el intento de transacción indicando que el Coste es mayor que el balance del cliente:

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

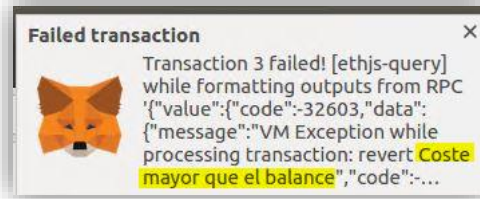


Figura 31. Transacción fallida: Cliente sin saldo FDT

Historia de Usuario 4 – Proveedor: consulta y ejecuta fumigaciones pendientes

El proveedor consulta las Fumigaciones reservadas pendientes de ejecución. Para ello, mediante la acción “Obtener reservas” obtiene una lista desplegable con todas las reservas pendientes de ejecución. Se selecciona una y se ejecuta mediante la acción “Fumigar”. Esto conlleva la liberación del dron y de la parcela para futuras reservas. Esta acción conlleva una transacción con su coste de gas correspondiente

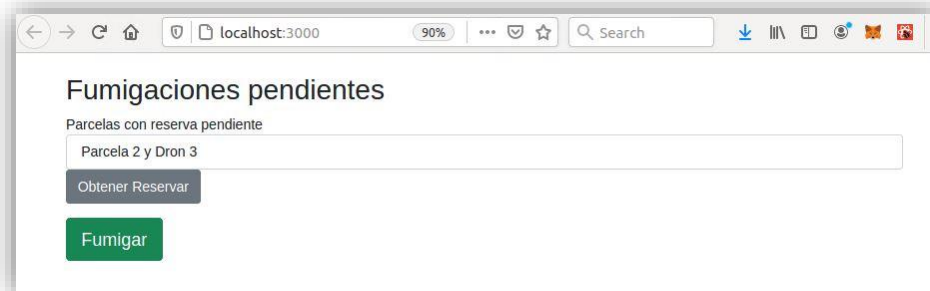


Figura 32. Interfaz web: Fumigar

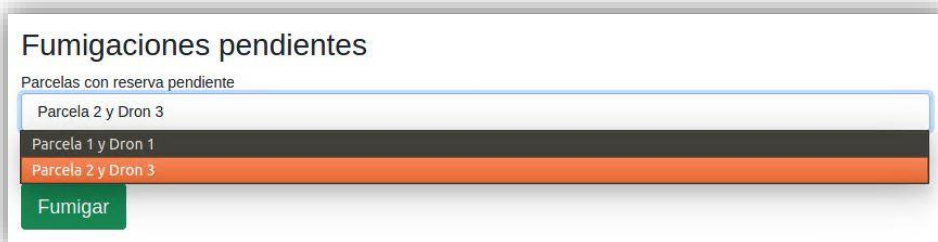


Figura 33. Interfaz web: Fumigar-valores parcelas/drones con reserva pendiente

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

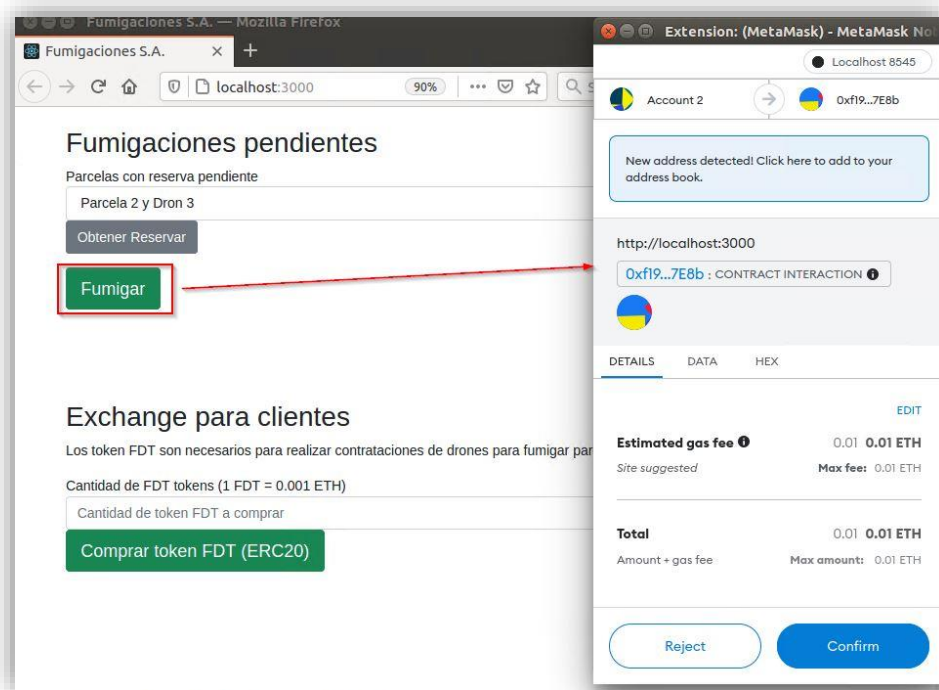


Figura 34. Interfaz web: Fumigar- transacción Metamask

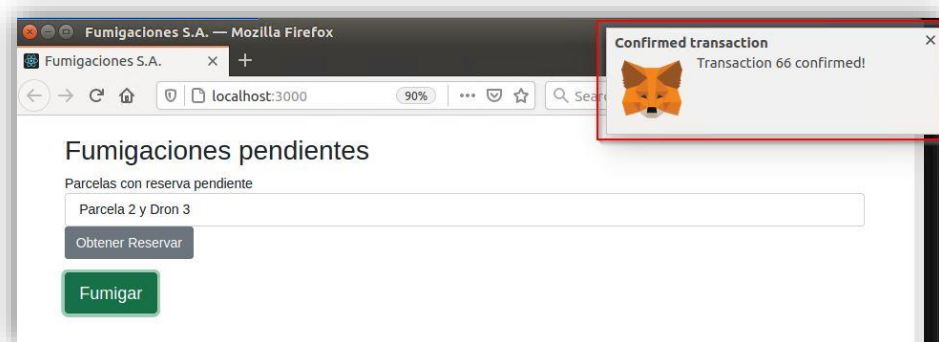


Figura 35. Interfaz web: Fumigar-confirmación transacción

En caso de que un cliente intente ejecutar una reserva, es decir, Fumigar, el sistema revertirá el intento de transacción indicando que el usuario no es el dueño, como se muestra a continuación:

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

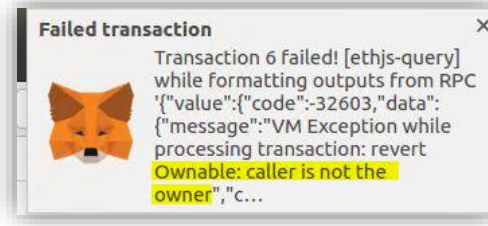


Figura 36. Interfaz web: Transacción fallida-Cliente intenta ejecutar una fumigación

Historia de Usuario 5 – Cliente: usar Exchange para obtener token FDT (ERC20)

Cliente puede obtener tokens, de la plataforma, que permiten la contratación de drones para fumigar una parcela. Para ello, puede indicar una cantidad de tokens FDT y ejecutar la acción “Comprar token FDT (ERC20)”. Conlleva la transferencia de la cantidad de tokens indicados, de la cuenta del contrato a la del cliente. Variando los saldos de ambas cuentas. También, es necesario realizar un pago en Eth relativo al precio de los tokens a comprar a razón de 1 FDT = 0.001 ETH. Y además, lleva el coste de gas correspondiente.

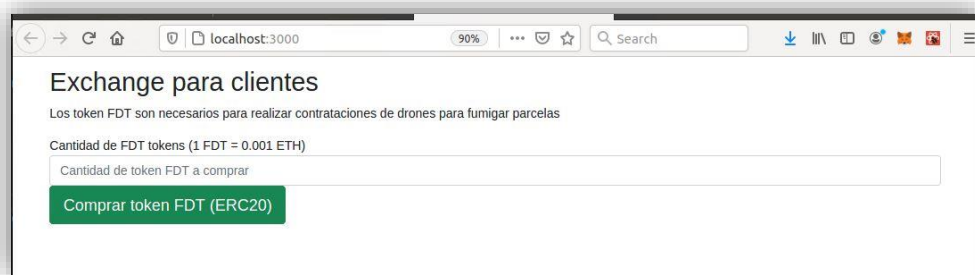


Figura 37. Interfaz web: Comprar token FDT

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

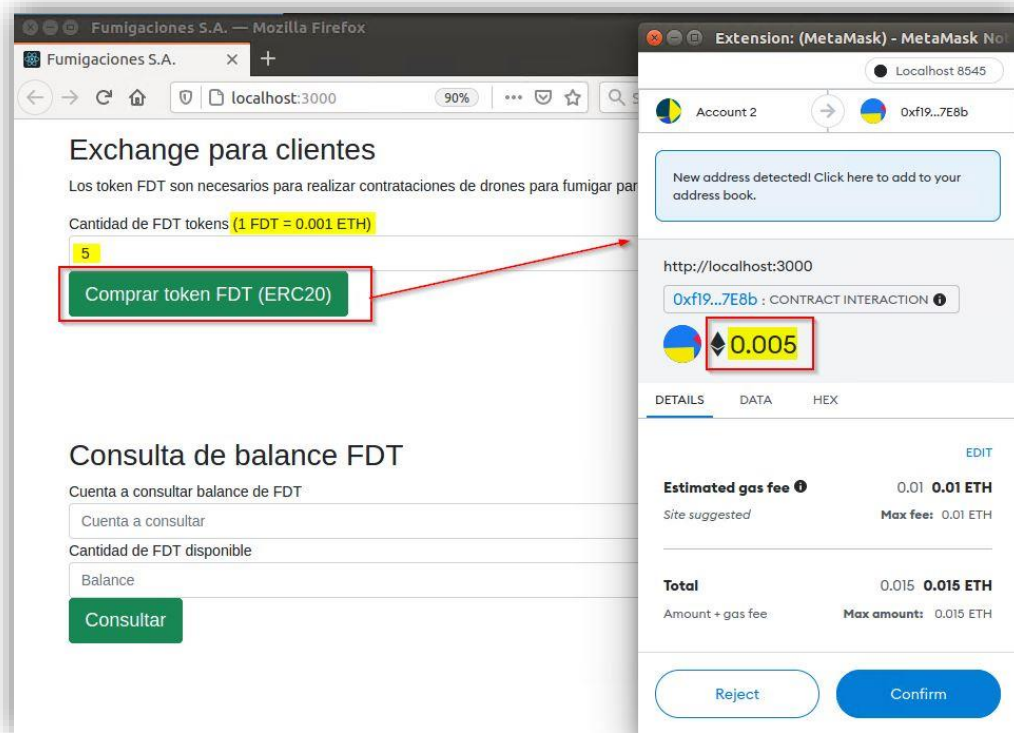


Figura 38. Interfaz web: Comprar token FDT- transacción Metamask

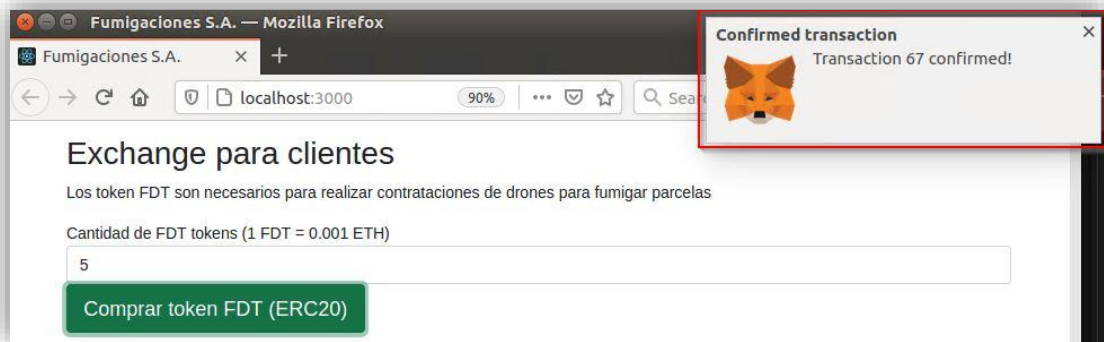


Figura 39. Interfaz web: Comprar token FDT- confirmación transacción

Ejemplo de movimiento de balances, donde:

Cuenta Cliente:0xffcf8fdee72ac11b5c542428b35eef5769c409f0

Cuenta Contrato:0x26b4AFb60d6C903165150C6F0AA14F8016bE4aec

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Consulta de balance FDT

Cuenta a consultar balance de FDT

Cantidad de FDT disponible

Consultar

Figura 40. Interfaz web: movimiento balance, consultar saldo cliente antes

Consulta de balance FDT

Cuenta a consultar balance de FDT

Cantidad de FDT disponible

Consultar

Figura 41. Interfaz web: movimiento balance, consultar saldo contrato antes

A continuación, el cliente realiza una compra de tokens FDT:

The screenshot shows a web browser window with the URL `localhost:3000` and a page titled "Exchange para clientes". The page contains a form to purchase FDT tokens, with a red box around the input field containing the number "10". A red arrow points from this input field to the MetaMask extension window. The MetaMask extension window shows a confirmation screen for a contract interaction at `http://localhost:3000`. The screen displays the account address `0x26b...4aec` and the estimated gas fee of 0.01 ETH. The total amount to be paid is 0.02 ETH. The screen also shows a "Reject" button and a "Confirm" button.

Figura 42. Interfaz web: movimiento balance, comprar token FDT

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Se comprueba los balances resultantes de Cliente y propietario observándose la correcta actualización de los mismos. El cliente aumenta en 10 su balance y el propietario lo reduce en esos mismos 10 tokens:

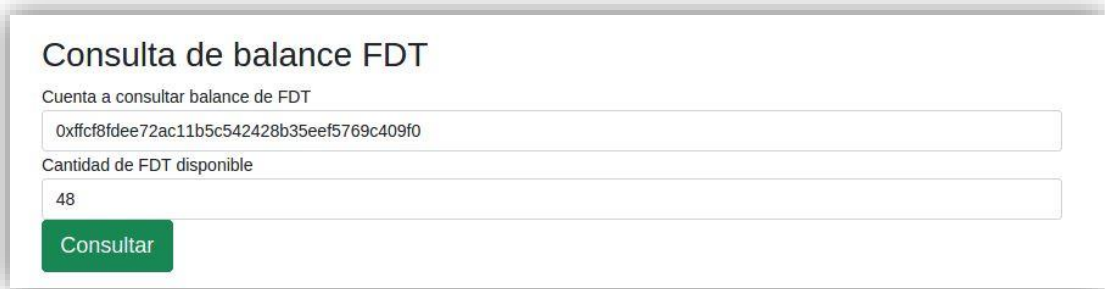


Figura 43. Interfaz web: movimiento balance, consultar saldo cliente después

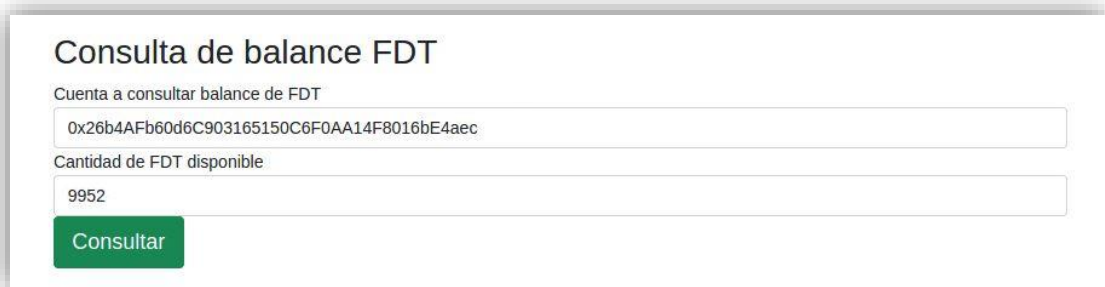


Figura 44. Interfaz web: movimiento balance, consultar saldo contrato después

Historia de Usuario 6 – Cliente: consultar saldo FDT

El cliente puede consultar en cualquier momento el saldo de una cuenta. Para ello debe informar la cuenta en el campo “Cuenta a consultar balance de FDT” y ejecutar la acción “Consultar”. A continuación, se muestra el saldo en el campo de “Cantidad de FDT disponible”. Esta acción no conlleva ningún gasto de gas.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Figura 45. Interfaz web: consultar saldo cuenta

Herramientas de ayuda

Se ha implementado un conjunto de funcionalidades que ayudan a comprobar y consultar los datos. Las funcionalidades están recogidas en la siguiente imagen:

Figura 46. Interfaz web: herramientas

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

La primera parte es para consultar datos de los drones. Todas ellas devuelven datos en la consola de desarrollador del navegador

- “Último Id dron” Devuelve el id del último dron registrado en la flota.
- Dron a mostrar: se indica previamente un id de dron para poder ejecutar las siguientes opciones
 - “Info dron”. Devuelve los datos que definen al dron
 - “Disponibilidad dron”: indica si está reservado para una fumigación. El valor devuelto es el índice del array de la fumigación reservada de manera que:
 - Un valor = -1 → disponible
 - Un valor ≥ 0 → no disponible
 - Coste dron: devuelve el coste en FDT de un dron por realizar una fumigación

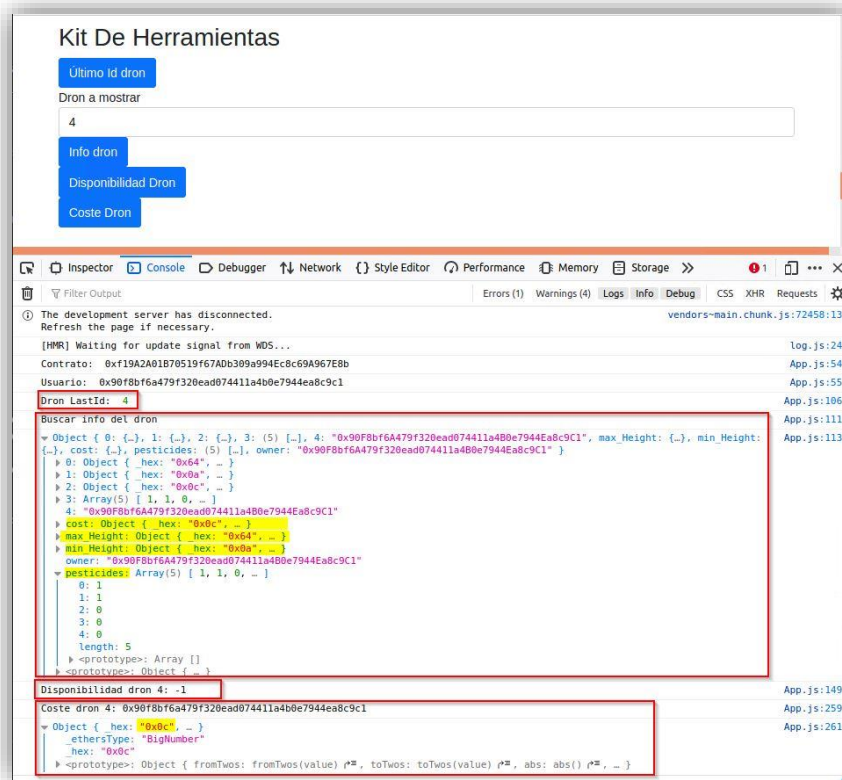


Figura 47. Interfaz web: herramientas- funcionalidad dron

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

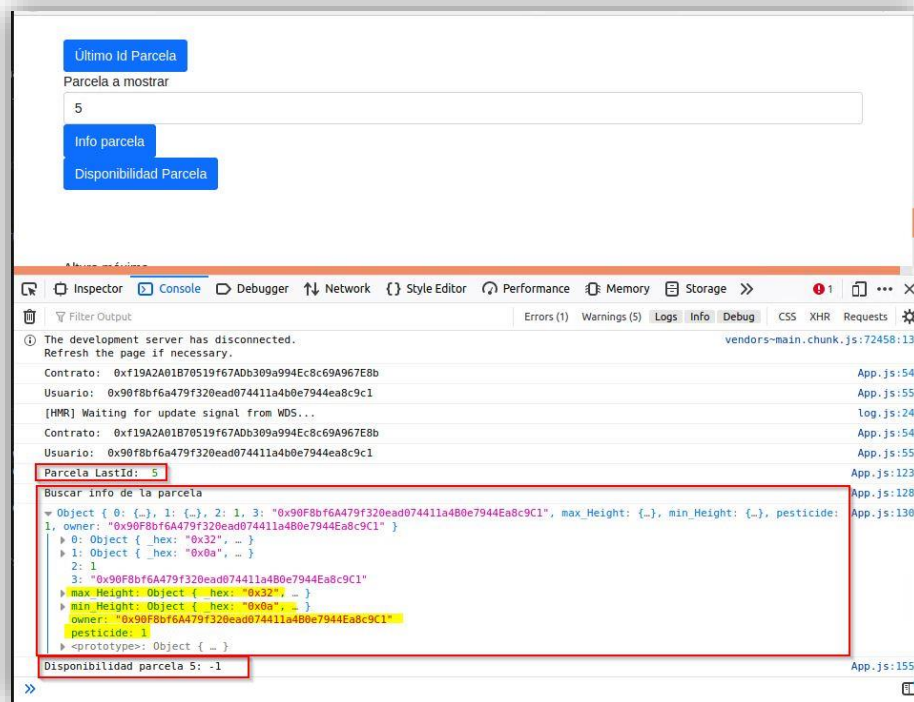


Figura 48. Interfaz web: herramientas- funcionalidad parcela

Para las parcelas, las acciones son equivalentes a sus homónimas en el apartado de herramientas sobre drones.

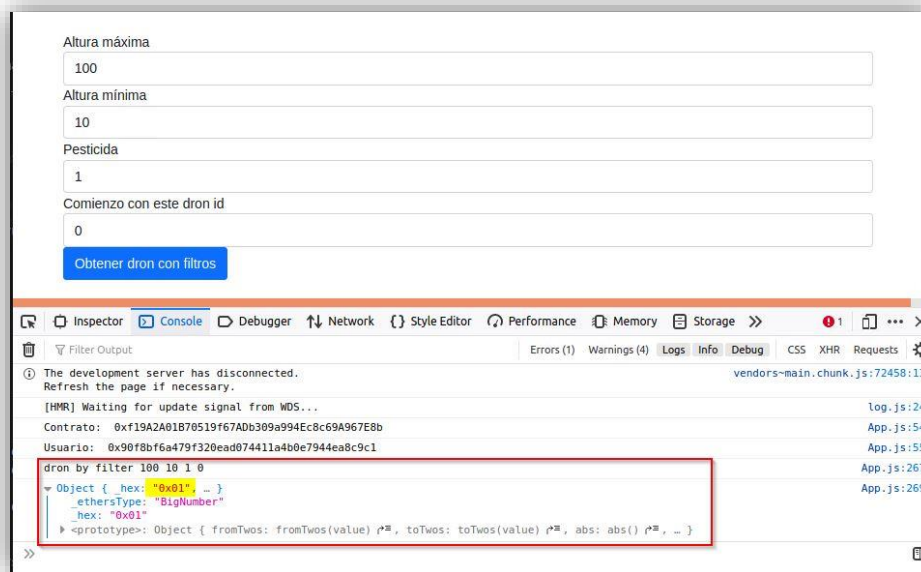


Figura 49. Interfaz web: herramientas- funcionalidad obtener dron confiltros

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

En este apartado, se intenta facilitar el seguimiento usado para asignar un dron a una parcela. Para ello, se informa una serie de filtros y se devuelve un dron disponible. De forma análoga al proceso “Buscar drones disponibles compatibles” del apartado de reserva de dron para la fumigación de una parcela.

A tener en cuenta, el campo “Comienzo con este dron id” es para buscar drones a partir de un id determinado, obviando todos aquellos con un id inferior.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Testing

A continuación, se va a realizar una batería de pruebas sobre alguna de las funcionalidades del proyecto de distinta complejidad. Para ello se va a utilizar otra de las características de la Truffle Suite, la herramienta de testing Truffle Test para realizar las pruebas unitarias:

Primero, se instalarán los componentes mocha y chai, de la siguiente manera:

```
$ npm install --save-dev mocha chai
```

A continuación, se crearan los ficheros de pruebas en un directorio llamado “test” en la raíz del proyecto. Estos tests están estructurados siguiendo la composición de las 3 A: Arrange o fase de preparación; Act o fase de acción; y Assert o fase de validación.

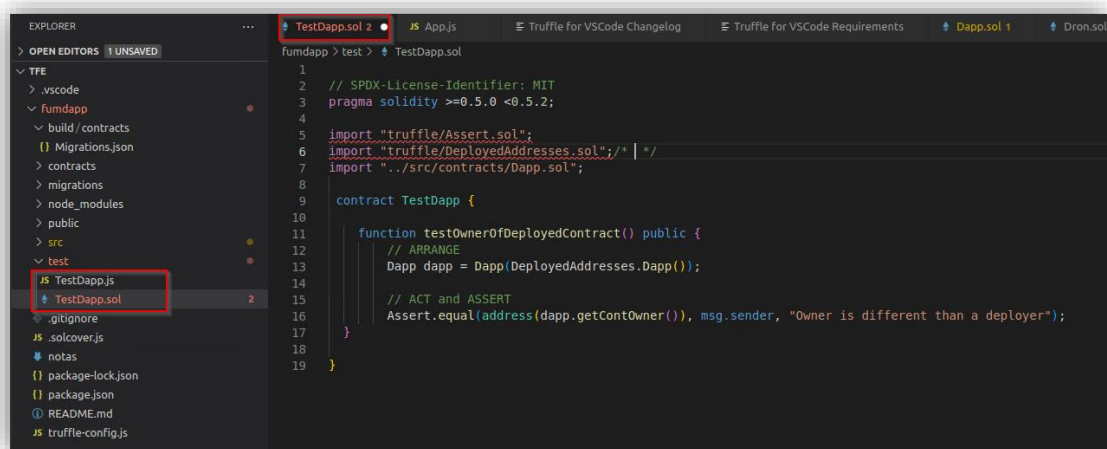


Figura 50. Testing: directorio test y fichero TestDapp.sol

En la imagen anterior se puede observar un ejemplo de pruebas del fichero “TestDapp.sol” (solidity) y a continuación otro ejemplo del fichero “TestApp.js” en javascript:

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

```

30
31 it("Customer balance after buy 5 FDT", async () => {
32
33     // El saldo inicial del contrato en 10000. El cliente comprará 5 FDT y pasarán a su cuenta
34
35     // ARRANGE
36     const dapp = await Dapp.deployed();
37     //const dapp = await Dapp.new({from: firstAccount});
38     const units_to_buy = 5;
39     const balance_DFT_customer_spected = 5;
40     const weis = web3.utils.toWei((units_to_buy*0.001).toString(), "ether");
41
42     // ACT
43     await dapp.buyTokensFDT(units_to_buy, {from: secondAccount, value: weis, gas: 500000});
44
45     const balance_DFT_customer = (
46         await dapp.getAccountBalance(secondAccount)
47     ).toNumber();
48
49
50     // ASSERT
51     assert.equal(balance_DFT_customer_spected, balance_DFT_customer);
52 });
53

```

Figura 51. Testing: fichero TestDapp.js

Una vez definidas las pruebas las ejecutamos de la siguiente manera:

\$ truffle test

```

usuario@ubuntu:~/tfe/fumdapp$ truffle test
Using network 'development'.

Compiling your contracts...
=====
> Compiling ./src/contracts/Dapp.sol
> Compiling ./test/TestDapp.sol

> compilation warnings encountered:

/home/usuario/tfe/fumdapp/src/contracts/Dapp.sol:3:1: Warning: Experimental features are turned on. Do not use experimental features on live deployments.
pragma experimental ABIEncoderV2;
^~~~~~

> Artifacts written to /tmp/test-12262-38782-9sqzuw.kv209
> Compiled successfully using:
   - solc: 0.5.1+commit.c8a2cb62.Emscripten.clang

TestDapp
  ✓ testOwnerOfDeployedContract (81ms)

Contract: Dapp
  ✓ sets an owner (262ms)
  ✓ Initial customer FDT balance (50ms)
  ✓ Customer balance after buy 5 FDT (112ms)
  ✓ Create a drone (298ms)
  ✓ Create a plot (115ms)

6 passing (11s)

```

Figura 52. Testing: Resultado ejecución de pruebas

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Se puede observar que todas las pruebas se han pasado correctamente. Programar utilizando una batería de pruebas ayuda a comprobar de forma rápida y ágil la funcionalidad de un proyecto cuando se introduce algún cambio. Al realizar algunas de estas pruebas, como por ejemplo el test definido para comprobar el incremento del balance de FDT en la cuenta del cliente al comprar tokens FDT, se han seguido los principios de TDD mediante el criterio de Arrange, Act y Assert. Se preparó el test con la funcionalidad que el sistema debía superar, se implementó dicha funcionalidad y se lanzaron los test. Se siguió un bucle entre desarrollo y ejecución de test hasta que el resultado de los test fue satisfactorio. Con estos sencillos ejemplos, se asimilaron muy bien los conceptos y ventajas del desarrollo dirigido por test tanto para los test en solidity como los test en javascript.

Se intento usar una segunda herramienta, en este caso para pruebas de calidad, para comprobar la cobertura de los test, el Solidity-coverage. En este caso no se consiguió hacerlo funcionar correctamente. Posiblemente por configuración pero lo más probable, por lo investigado en el tema, por un incompatibilidad de versiones. Se configuró el fichero truffle-config.js, se creo un .solcover.js en al raíz del proyecto, se jugó con distintos parámetros de este último y se obtenía siempre el mismo resultado al ejecutarlo desde la raíz del producto

```
// Plugins
plugins: ["solidity-coverage"],
```

Figura 53. Configuración solidity-coverage en truffle-config.js

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

```

usuario@ubuntu:~/tfe/fumdapp$ truffle run coverage
WARNING: Missing strong random number source

> Using Truffle library from local node_modules.

> solidity-coverage cleaning up, shutting down ganache server
usuario@ubuntu:~/tfe/fumdapp$

```

Figura 54. Resultado obtenido de solidity-coverage

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Conclusiones

Mediante la realización de este proyecto, acotado por el alcance mencionado en el mismo, se ha conseguido asimilar y poner en práctica los conceptos teóricos vistos durante el curso. Además, las investigaciones continuas sobre las herramientas y paquetes utilizados han ayudado a contextualizar el papel de cada uno.

De igual manera, se han interiorizado las distintas formas que hay de desarrollar y desplegar los contratos en distintas redes, así como la manera de conectarse a redes mediante clientes, desbloqueo de cuentas, interacción con Smart Contracts desde interfaces web, realización de pagos, el funcionamiento de los tokens ERC721 y ERC20 y la realización de baterías de pruebas.

Fue un gran reto solventar las incompatibilidades de versiones entre los distintos componentes del proyecto, todo condicionado por la versión de Solidity elegida, mencionado en apartados anteriores. Dichos inconvenientes produjeron un retraso estimado en 10-12 días (a unas 4 semanas del fin del plazo de entrega del proyecto) que conllevó a realizar un gran sprint posterior mantenido hasta el final del proyecto para poder completar el mismo en fecha.

Las tres últimas semanas de proyecto fueron las más activas en cuanto a desarrollo debido a que parte del contenido del curso no se había visto aún en las clases y se empezó a aplicar en el proyecto a medida que se iban emitiendo las clases.

A dos semanas del fin del plazo, se empezó a percibir una ausencia por parte del “cliente” ya que no se recibían respuestas a las preguntas trasladadas, lo que dificultó bastante la progresión del proyecto.

Por todo ello, produce una gran satisfacción haber podido sobreponerse a las dificultades acaecidas por el camino y llegar a este punto haciendo uso de las cualidades resolutivas adquiridas como un gran ingeniero informático (y científico de datos), aunque el balance general supusiese una dedicación de recursos y de tiempo mucho mayor a lo planificado.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Como conclusión, haber conseguido realizar este proyecto y desplegarlo en Alastria con todos los procesos intermedios recogidos en este documento se percibe como un objetivo cumplido, visto como un punto de partida en el mundo blockchain para continuar progresivamente el desarrollo profesional en el mismo.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Estructura del proyecto

A continuación, se detallará como está estructurado el código del proyecto

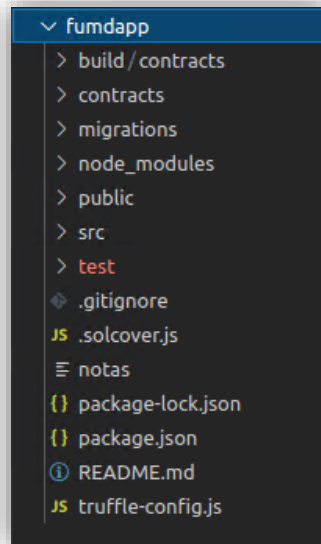


Figura 55. Estructura general del proyecto (directorio raíz)

En el directorio raíz se encuentra tanto el directorio “test” (pruebas) como los ficheros de configuración. Ejemplo: package.json y truffle-config.js

También, se encuentra el directorio “src”, que contiene el código:

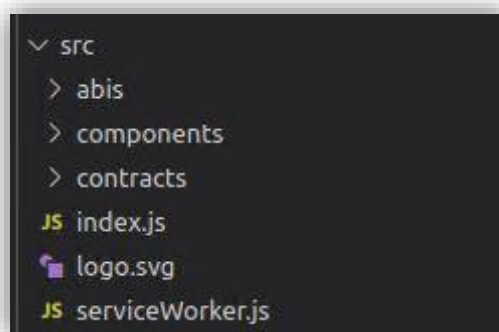


Figura 56. Estructura: directorio src

Podemos ver que aquí se encuentra:

- abis: los abis, en formato .json, generados
- components: directorio con los ficheros necesarios para en interfaz web por medio de React (que también utiliza los ficheros index.js y serviceWorker.js situados en src), Bootstrap, Popper.js y jquery.

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

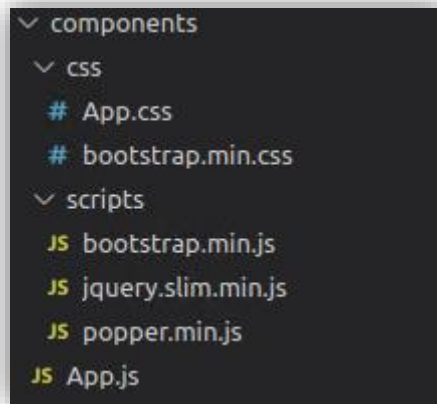


Figura 57. Estructura: directorio components

- contracts: directorio con los contratos inteligentes del proyecto. En él se encuentran tanto los creados para este proyecto como los importados de otras librerías (por ejemplo, openzeppelin). Se han juntado, los contratos importados, en el mismo directorio para facilitar su despliegue con otras herramientas como remix y, quizás, facilitar así la tarea de corrección/comprobación de este proyecto. En la siguiente imagen se pueden ver todos los contratos y, concretamente, remarcados aquellos creados para este proyecto:

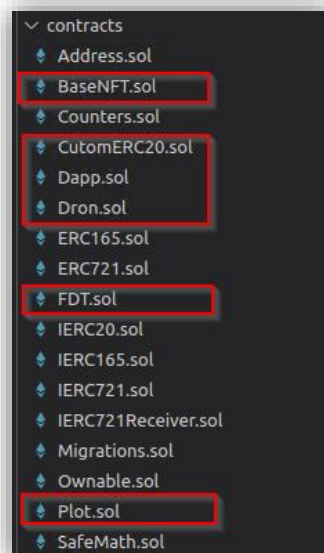


Figura 58. Estructura: directortio contracts resaltando los contratos desarrollados

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Como se mencionaba en el apartado de testing, hay un directorio “test” en la raíz del proyecto donde incluir los ficheros con el conjunto de las pruebas. En este caso tenemos las pruebas realizadas con solidity y con javascript:

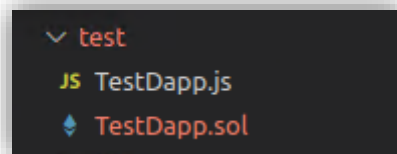


Figura 59. Estructura: directorio test (raíz del proyecto)

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

Índice de figuras

FIGURA 1.	DIAGRAMA DE DESPLIEGUE.....	12
FIGURA 2.	DIAGRAMA SECUENCIA - REGISTRAR DRON	13
FIGURA 3.	DIAGRAMA SECUENCIA - REGISTRAR PARCELA.....	13
FIGURA 4.	DIAGRAMA SECUENCIA – RESERVAR FUMIGACIÓN	14
FIGURA 5.	DIAGRAMA SECUENCIA - FUMIGAR	14
FIGURA 6.	DIAGRAMA SECUENCIA – COMPRAR TOKEN	15
FIGURA 7.	DIAGRAMA SECUENCIA – CONSULTAR SALDO	15
FIGURA 8.	CUENTAS DISPONIBLES.....	17
FIGURA 9.	VERSIÓN DEL COMPILADOR	18
FIGURA 10.	VALORES DE ENVIRONMENT Y ACCOUNT	19
FIGURA 11.	ENDPOINT REMIX, CONEXIÓN CON ALASTRIA	19
FIGURA 12.	CONTRATO DESPLEGADO	20
FIGURA 13.	EJECUCIÓN DE BUYTOKENFDT	21
FIGURA 14.	LOG EJECUCIÓN BUYTOKEN FDT	21
FIGURA 15.	EJECUCIÓN GETDRONINFO	22
FIGURA 16.	AVISO AL COMPILAR	22
FIGURA 17.	CONJUNTO DE HERRAMIENTAS DE DESARROLLO (ELABORACIÓN PROPIA)	24
FIGURA 18.	VERSIÓN DE TRUFFLE	25
FIGURA 19.	VALORES DE SCRIPTS EN FICHERO PACKAGE.JSON	26
FIGURA 20.	INTERFAZ WEB: REGISTRO DRON	28
FIGURA 21.	INTERFAZ WEB: REGISTRO DRON-TRANSACCIÓN METAMASK.....	29
FIGURA 22.	INTERFAZ WEB: REGISTRO DRON-CONFIRMACIÓN TRANSACCIÓN	29
FIGURA 23.	INTERFAZ WEB: REGISTRO PARCELA.....	30
FIGURA 24.	INTERFAZ WEB: REGISTRO PARCELA-VALORES PESTICIDA ACEPTADO	30
FIGURA 25.	INTERFAZ WEB: REGISTRO PARCELA-TRANSACCIÓN METAMASK	31

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

FIGURA 26.	INTERFAZ WEB: REGISTRO PARCELA-CONFIRMACIÓN TRANSACCIÓN	31
FIGURA 27.	INTERFAZ WEB: RESERVAR	32
FIGURA 28.	INTERFAZ WEB: RESERVAR-VALORES PARCELA	32
FIGURA 29.	INTERFAZ WEB: RESERVAR – TRANSACCIÓN METAMASK.....	33
FIGURA 30.	INTERFAZ WEB: RESERVAR-CONFIRMACIÓN TRANSACCIÓN	33
FIGURA 31.	TRANSACCIÓN FALLIDA: CLIENTE SIN SALDO FDT	34
FIGURA 32.	INTERFAZ WEB: FUMIGAR	34
FIGURA 33.	INTERFAZ WEB: FUMIGAR-VALORES PARCELAS/DRONES CON RESERVA PENDIENTE 34	
FIGURA 34.	INTERFAZ WEB: FUMIGAR- TRANSACCIÓN METAMASK.....	35
FIGURA 35.	INTERFAZ WEB: FUMIGAR-CONFIRMACIÓN TRANSACCIÓN.....	35
FIGURA 36.	INTERFAZ WEB: TRANSACCIÓN FALLIDA-CLIENTE INTENTA EJECUTAR UNA FUMIGACIÓN 36	
FIGURA 37.	INTERFAZ WEB: COMPRAR TOKEN FDT.....	36
FIGURA 38.	INTERFAZ WEB: COMPRAR TOKEN FDT- TRANSACCIÓN METAMASK.....	37
FIGURA 39.	INTERFAZ WEB: COMPRAR TOKEN FDT- CONFIRMACIÓN TRANSACCIÓN	37
FIGURA 40.	INTERFAZ WEB: MOVIMIENTO BALANCE, CONSULTAR SALDO CLIENTE ANTES.....	38
FIGURA 41.	INTERFAZ WEB: MOVIMIENTO BALANCE, CONSULTAR SALDO CONTRATO ANTES ..	38
FIGURA 42.	INTERFAZ WEB: MOVIMIENTO BALANCE, COMPRAR TOKEN FDT	38
FIGURA 43.	INTERFAZ WEB: MOVIMIENTO BALANCE, CONSULTAR SALDO CLIENTE DESPUÉS....	39
FIGURA 44.	INTERFAZ WEB: MOVIMIENTO BALANCE, CONSULTAR SALDO CONTRATO DESPUÉS 39	
FIGURA 45.	INTERFAZ WEB: CONSULTAR SALDO CUENTA	40
FIGURA 46.	INTERFAZ WEB: HERRAMIENTAS	40
FIGURA 47.	INTERFAZ WEB: HERRAMIENTAS- FUNCIONALIDAD DRON	41
FIGURA 48.	INTERFAZ WEB: HERRAMIENTAS- FUNCIONALIDAD PARCELA	42
FIGURA 49.	INTERFAZ WEB: HERRAMIENTAS- FUNCIONALIDAD OBTENER DRON CONFILTROS..	42

Asignatura	Datos del alumno	Fecha
Trabajo Final	Apellidos: Martínez Arellano	04/07/2022
	Nombre: Javier	

FIGURA 50.	TESTING: DIRECTORIO TEST Y FICHERO TESTDAPP.SOL.....	44
FIGURA 51.	TESTING: FICHERO TESTDAPP.JS	45
FIGURA 52.	TESTING: RESULTADO EJECUCIÓN DE PRUEBAS	45
FIGURA 53.	CONFIGURACIÓN SOLIDITY-COVERAGE EN TRUFFLE-CONFIG.JS.....	46
FIGURA 54.	RESULTADO OBTENIDO DE SOLIDITY-COVERAGE.....	47
FIGURA 55.	ESTRUCTURA GENERAL DEL PROYECTO (DIRECTORIO RAÍZ)	50
FIGURA 56.	ESTRUCTURA: DIRECTORIO SRC.....	50
FIGURA 57.	ESTRUCTURA: DIRECTORIO COMPONENTS.....	51
FIGURA 58.	ESTRUCTURA: DIRECTORTIO CONTRACTS RESALTANDO LOS CONTRATOS DESARROLLADOS	51
FIGURA 59.	ESTRUCTURA: DIRECTORIO TEST (RAÍZ DEL PROYECTO)	52

Índice de tablas

TABLA 1.	ROLES	6
TABLA 2.	HISTORIAS DE USUARIO	10