SMART CONTRACT EN BLOCKCHAIN EN RED LOCAL Y RED PÚBLICA DE PRUEBAS

RUBEN ENRIQUE CAÑON

YEISON EFRAIN TAFUR

HARVEY RICARDO PEÑALOZA

PROFESOR

ELVIS EDUARDO GAONA GARCIA

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE INGENIERÍA

MAESTRÍA EN CIENCIAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

BLOCKCHAIN

JUNIO

Contenido

[Tabla de Ilustraciones 3](#_Toc12484576)

[1. Motivación del proyecto 4](#_Toc12484577)

[2. Estructura general de la solución 4](#_Toc12484578)

[3. Entorno de implementación 6](#_Toc12484579)

[3.1 Instalación de NodeJS. 7](#_Toc12484580)

[3.2 Instalación y configuración de proyecto Truffle (ambiente de desarrollo de Dapps ) 9](#_Toc12484581)

[3.3 Instalación de Ganache 9](#_Toc12484582)

[4. Entorno de aplicación local 10](#_Toc12484583)

[5. Entorno de aplicación Red Ropsten 14](#_Toc12484584)

[6. Código fuente aplicación 23](#_Toc12484585)

# Tabla de Ilustraciones

[Ilustración 1 Arquitectura de aplicación local 5](#_Toc12484614)

[Ilustración 2 Arquitectura de aplicación sobre red publica ropsten 6](#_Toc12484615)

[Ilustración 3 Instalación Node 7](#_Toc12484616)

[Ilustración 4 Aceptación de términos de licencia Node 7](#_Toc12484617)

[Ilustración 5 Iniciar instalación node 8](#_Toc12484618)

[Ilustración 6 Instalación de node iniciada 8](#_Toc12484619)

[Ilustración 7 Finalización de instalación de node 9](#_Toc12484620)

[Ilustración 8 Aplicación Ganache 10](#_Toc12484621)

[Ilustración 9 Ingreso a la aplicación 10](#_Toc12484622)

[Ilustración 10 Administración de usuarios 11](#_Toc12484623)

[Ilustración 11 Creación de usuarios 11](#_Toc12484624)

[Ilustración 12 Creacion de usuario, confirmación en MetaMask 12](#_Toc12484625)

[Ilustración 13 Creación de usuario exitosa 12](#_Toc12484626)

[Ilustración 14 Asignación de roles 13](#_Toc12484627)

[Ilustración 15 Asignación de rol a usuario confirmar la transacción 13](#_Toc12484628)

[Ilustración 16 Instalacion HDWalletProvider 14](#_Toc12484629)

[Ilustración 17 Importar Red Privada Ropsten 15](#_Toc12484630)

[Ilustración 18 Carga de ETH en red pública 16](#_Toc12484631)

[Ilustración 19 Despliegue de contrato en red ropsten 16](#_Toc12484632)

[Ilustración 20 Ejecución exitosa de contrato en red Ropsten 17](#_Toc12484633)

[Ilustración 21 Visualización de transacciones sobre red ropsten 18](#_Toc12484634)

[Ilustración 22 Confirmación de transacción sobre ropsten 19](#_Toc12484635)

[Ilustración 23 Validación de transacción ropsten 20](#_Toc12484636)

[Ilustración 24 Resumen de transacción en red ropsten 21](#_Toc12484637)

[Ilustración 25 Transacción confirmada 21](#_Toc12484638)

[Ilustración 26 Descripción de transacción 22](#_Toc12484639)

[Ilustración 27 Vista de Contrato desplegado en red ropsten 23](#_Toc12484640)

# Motivación del proyecto

El control de acceso consiste en un mecanismo que permite verificar la identidad de un usuario u ordenador con el fin de autorizar el ingreso o acceso a recursos físicos o lógicos.[[1]](#footnote-1) En implementaciones típicas de un sistema de control de acceso, las credenciales de un usuario u ordenador estarían centralizadas, esto tiene la desventaja de que en caso de fallo la disponibilidad de acceso se vería afectada, también su integridad podría comprometerse en caso de que un atacante pudiese acceder a la fuente de datos centralizada que determina a que recursos un usuario u ordenador puede o no consentir.

Blockchain, entre otras funcionalidades, puede compararse con “(…) un gigantesco libro de cuentas en los que los registros (los bloques) están enlazados y cifrados para proteger la seguridad y privacidad de las transacciones. Es, en otras palabras, una base de datos distribuida y segura (gracias al cifrado) que se puede aplicar a todo tipo de transacciones que no tienen por qué ser necesariamente económicas.”[[2]](#footnote-2).

Al usar una cadena de bloques para el control de acceso, se lograría tanto distribuir las cuentas y permisos entre los nodos que pertenecen a la red blockchain, como garantizar su seguridad gracias al cifrado que implementa dicha tecnología y evitar tanto la indisponibilidad como la integridad de las credenciales de los usuarios u ordenadores.

En el presente documento se describen las tecnologías utilizadas para implementar un contrato inteligente que provee un sistema de control de acceso de usuarios. Dicha implementación fue alojada en una red local como en una red pública de pruebas.

# Estructura general de la solución

Para la implementación del contrato inteligente, se hizo uso del framework “Truffle” para el desarrollo del Smart Contract. Truffle[[3]](#footnote-3) permite, entre otras cosas, la compilación de contratos desarrollados en Solidity y el despliegue del mismo en redes locales o públicas., así como la ejecución de pruebas sobre los contratos que allí se desarrollen.

Se mencionó anteriormente, que la implementación del contrato inteligente de control de acceso se desplegó en una red local y en una pública de pruebas. Para la red local se utilizó la herramienta “Ganache”[[4]](#footnote-4), esta es una red de nodos de blobkchain local para el desarrollo en Etehereum, en ella se pueden desplegar contratos como el desarrollado en el presente proyecto.

En el siguiente gráfico se muestran los componentes utilizados para la implementación y ejecución del control de acceso en blockchain para una red local:

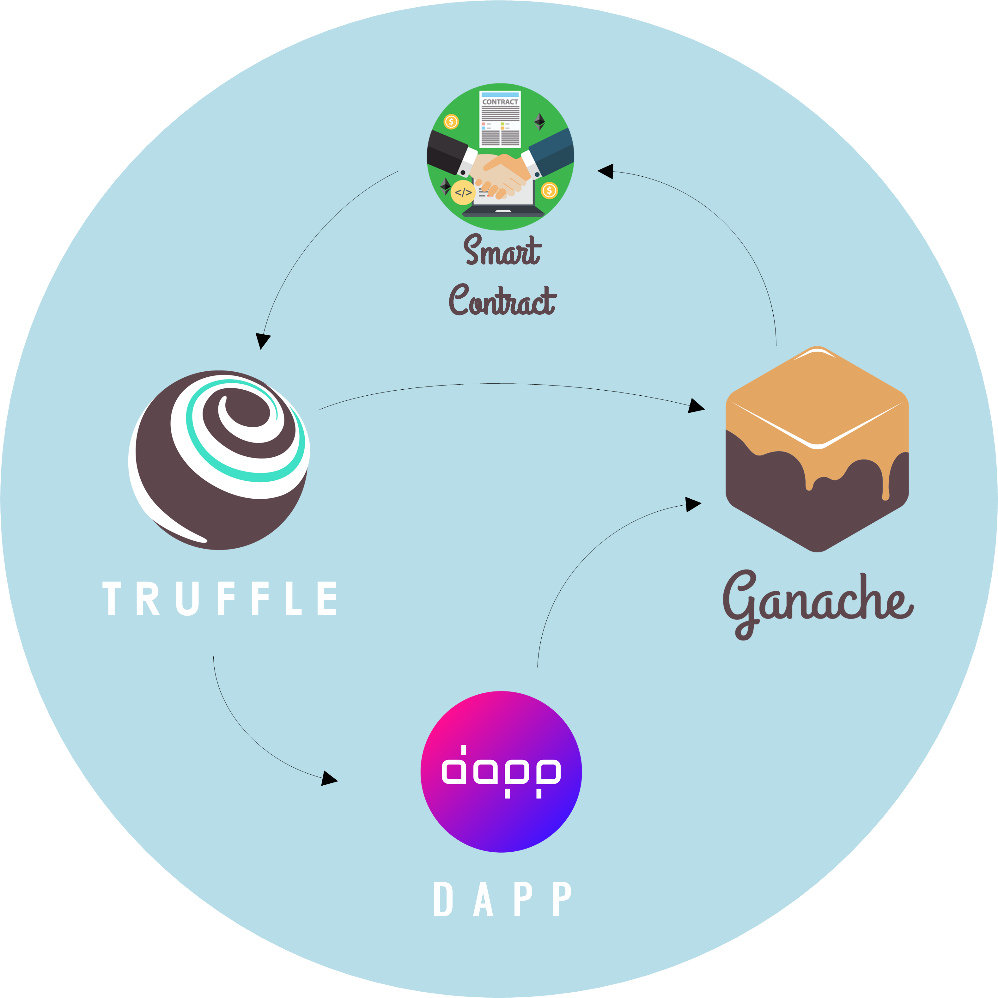


Ilustración 1 Arquitectura de aplicación local

* El Smart contract se desarrolla en Truffle,
* Este se despliega en la red local Ganache, así la red local puede hacer uso del contrato
* Se usa una DAPP, que “(…) es una aplicación que utiliza blockchain como base de datos o como núcleo donde implementar su lógica de negocio.”[[5]](#footnote-5). Esta DAPP se comunica con ganache para usar el contrato inteligente y así poder verificar las funcionalidades del contrato de control de acceso.

Para la implementación del contrato sobre la red pública de pruebas se hizo uso de la red Ropsten, la estructura de la solución es igual que la anteriormente descrita, salvo que la red blockhain Ganache es reemplazada por Ropsten:

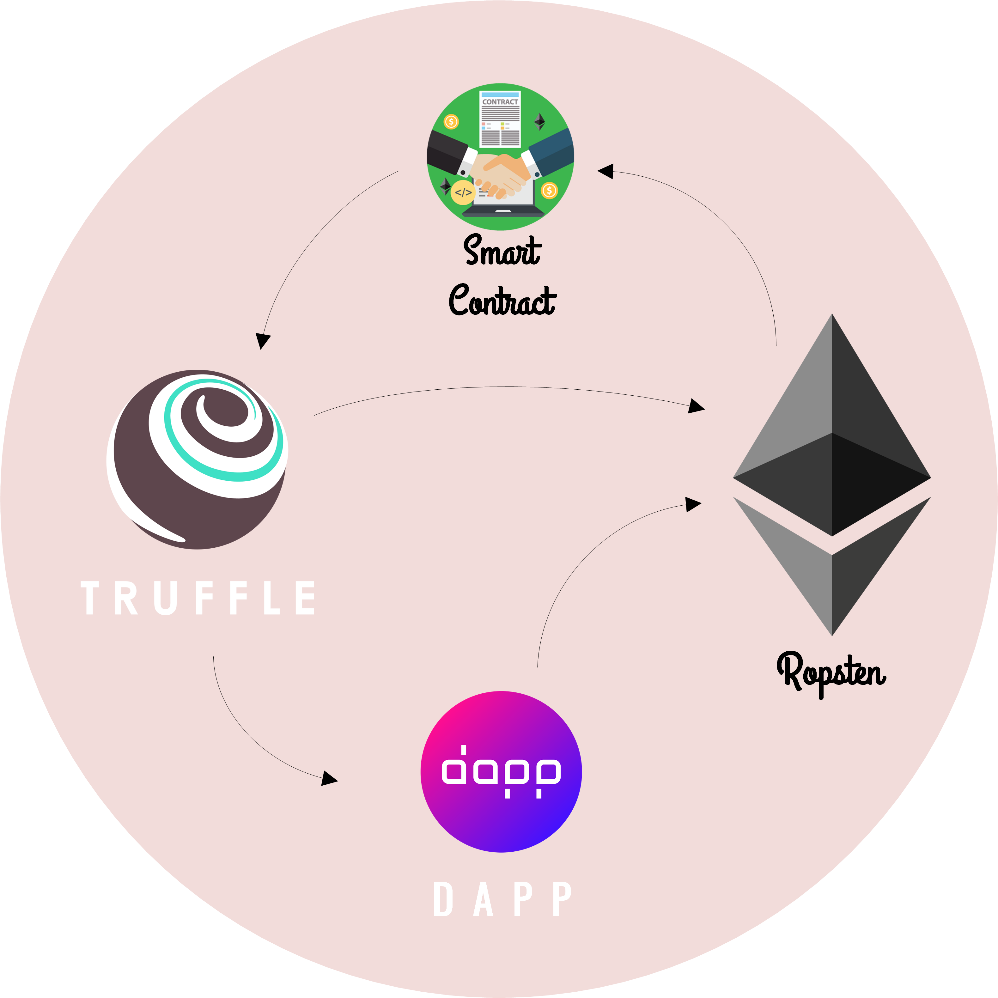


Ilustración 2 Arquitectura de aplicación sobre red publica ropsten

# Entorno de implementación

Para el alistamiento del entorno de desarrollo en el cual se implementó la red, se hizo uso de NodeJS, este puede descargarse directamente desde la página del fabricante <https://nodejs.org/es/download/>, se encuentra disponible para diferentes plataformas, la usada fue la soportada por Windows.

## Instalación de NodeJS.

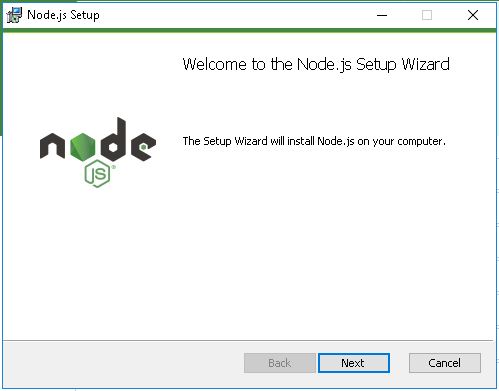


Ilustración 3 Instalación Node



Ilustración 4 Aceptación de términos de licencia Node

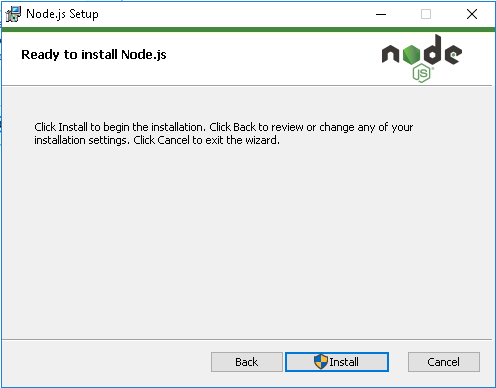


Ilustración 5 Iniciar instalación node

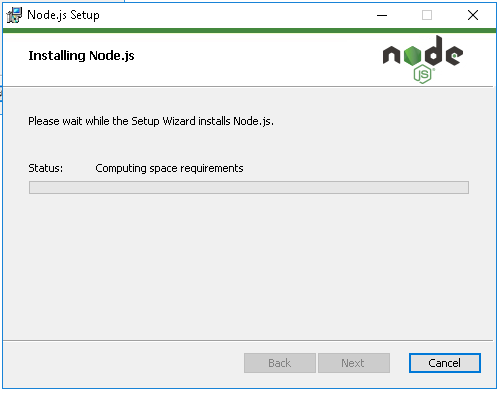


Ilustración 6 Instalación de node iniciada

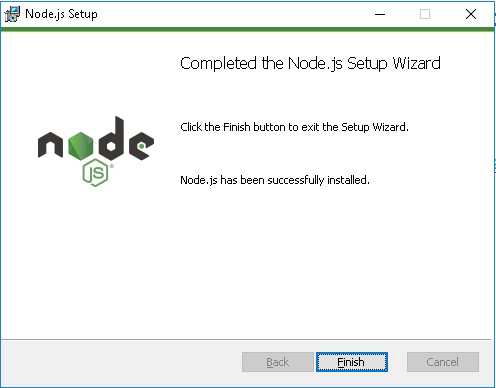


Ilustración 7 Finalización de instalación de node

## Instalación y configuración de proyecto Truffle (ambiente de desarrollo de Dapps )

Ejecutar desde consola de comandos:

* npm install -g truffle #instalación de ambiente de desarrollo de Dapps
* mkdir controlAcceso #creación de carpeta del proyecto.
* cd controlAcceso
* truffle unbox webpack #Creaciòn de plantilla de proyecto web.

## Instalación de Ganache

Puede descargarse directamente desde la página del fabricante en este enlace: <https://www.trufflesuite.com/ganache>; o a través de la línea de comandos usando el manejador de paquetes de node:

* npm install -g ganache-cli # instalación de cliente ganache (blockchain local)

Al ejecutar ganache se despliega en la interfaz las cuentas configuradas en la cadena de bloques local:

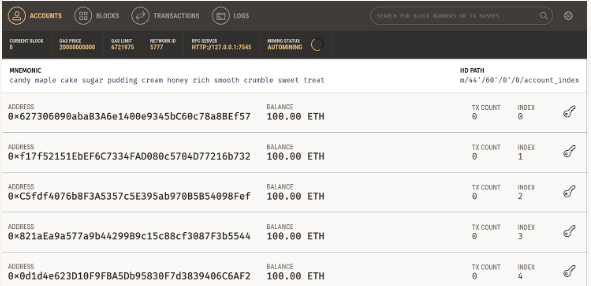


Ilustración 8 Aplicación Ganache

# Entorno de aplicación local

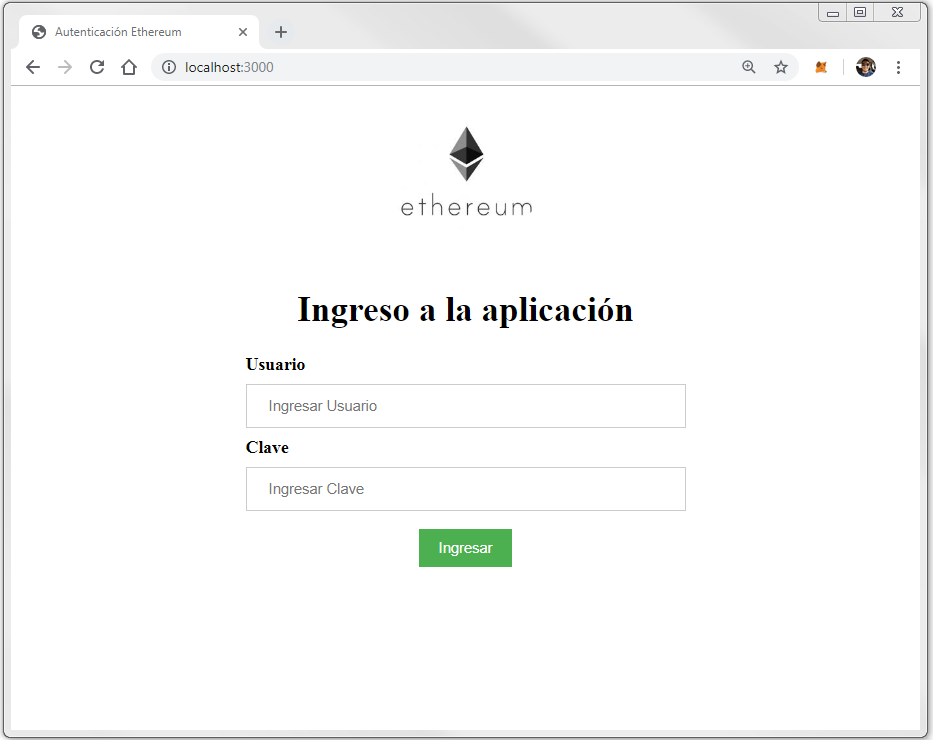


Ilustración 9 Ingreso a la aplicación

Pantalla de inicio de sesión de la aplicación, para su correcto funcionamiento debe estar ***Ganache*** activo y ***Truffle*** en estado migrado.

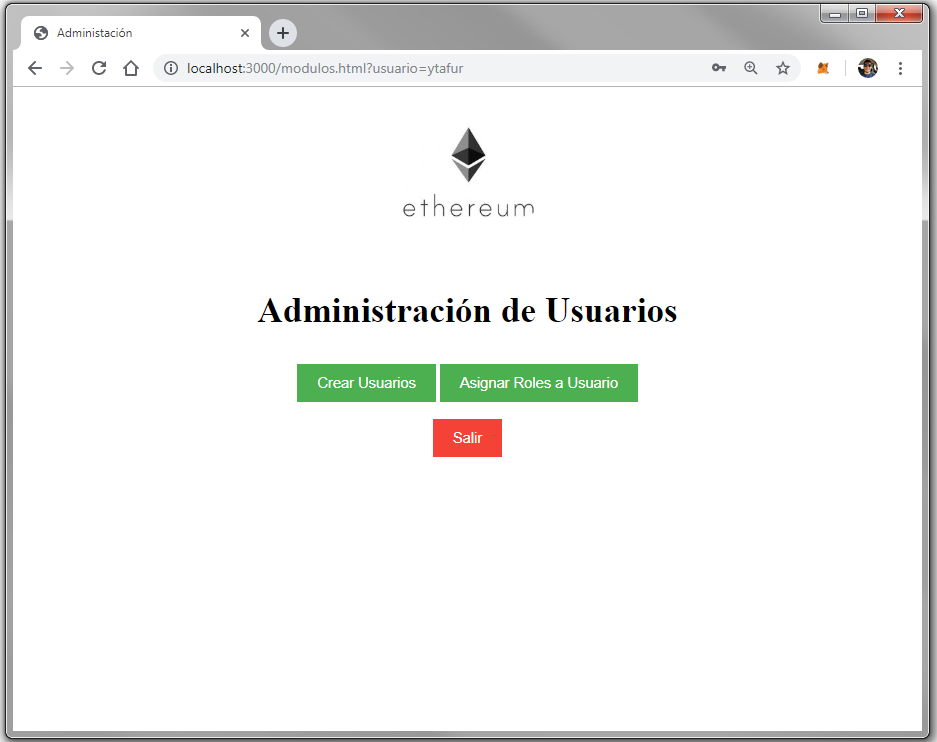


Ilustración 10 Administración de usuarios

Al iniciar sesión se emplea el Smart contract, este realiza la validación de usuario y clave permitiendo el acceso a la página de administración de usuarios, en esta página se tienen las siguientes dos opciones de actividad

* Crear Usuarios
* Asignar roles a usuarios

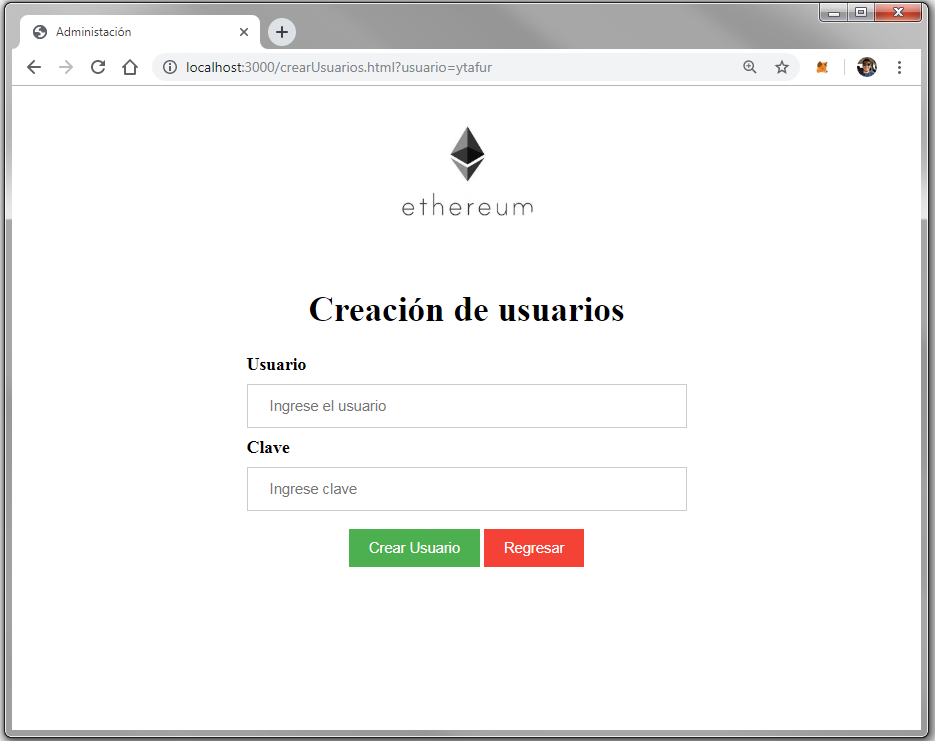


Ilustración 11 Creación de usuarios

Para poder realizar la creación de usuarios es necesario tener asignado el rol “**superadmin**”, en caso de no contrar con el rol, se despliega mensaje indicando que no se tienen los privilegios necesarios.

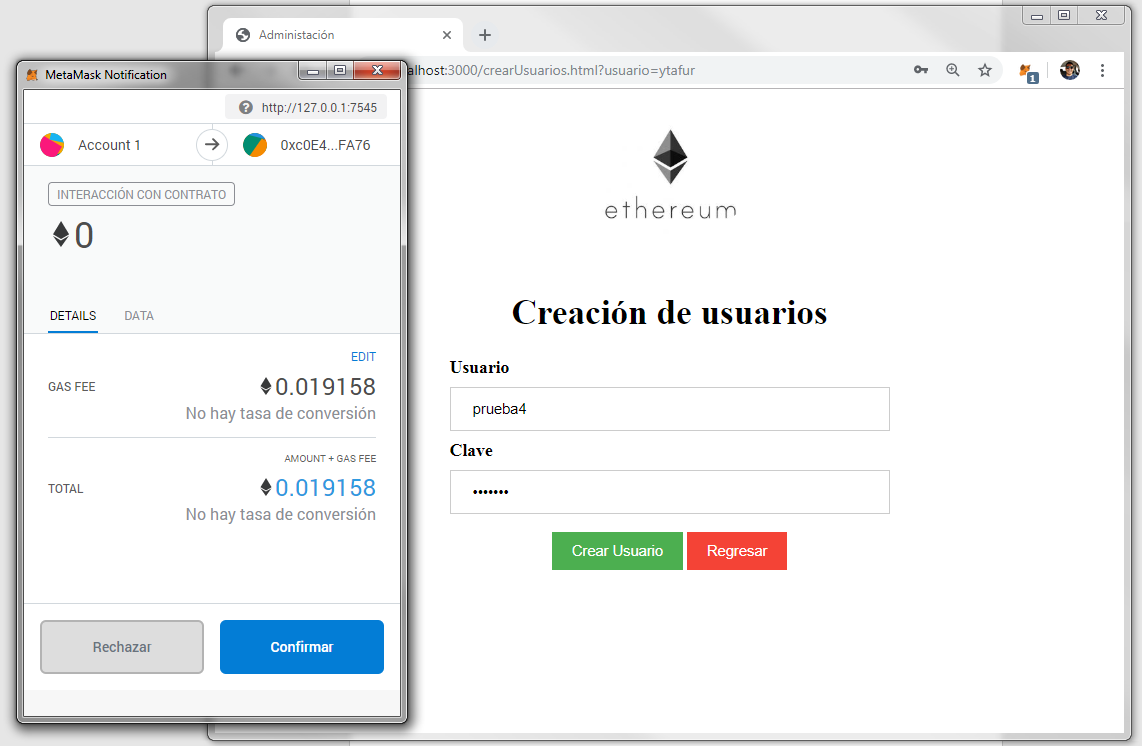


Ilustración 12 Creacion de usuario, confirmación en MetaMask

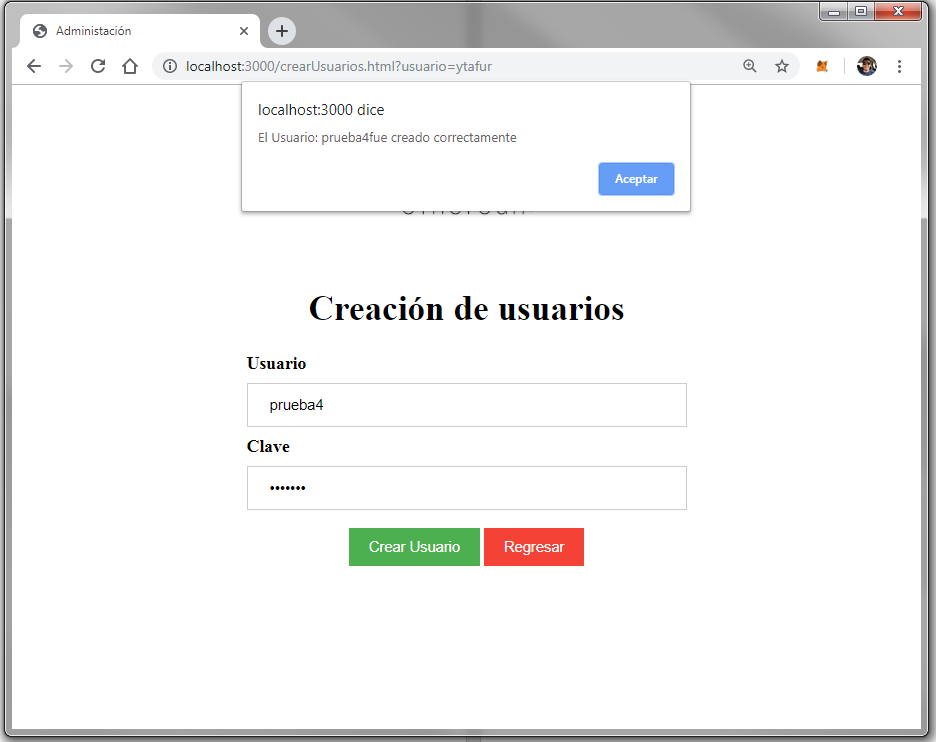


Ilustración 13 Creación de usuario exitosa

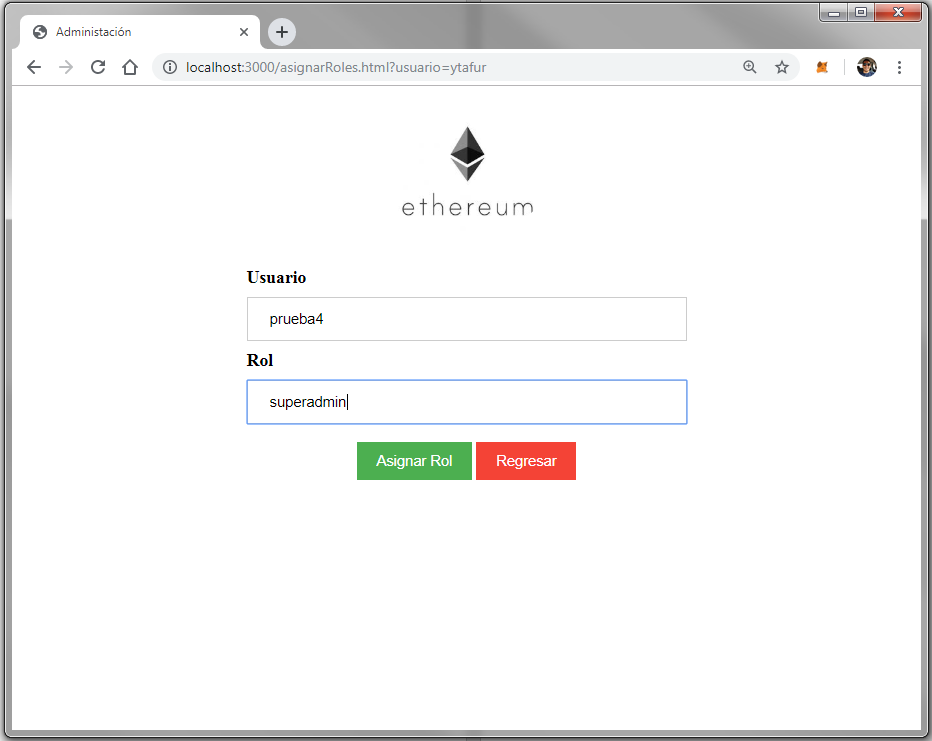


Ilustración 14 Asignación de roles

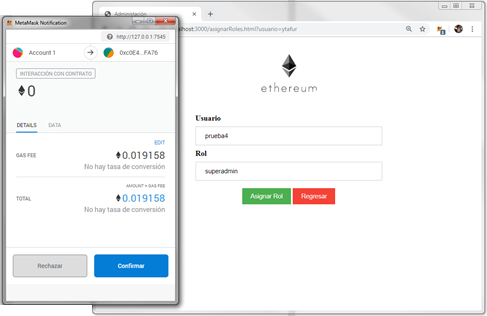


Ilustración 15 Asignación de rol a usuario confirmar la transacción

# Entorno de aplicación Red Ropsten

* 1. Instalacion de HDWalletProvider

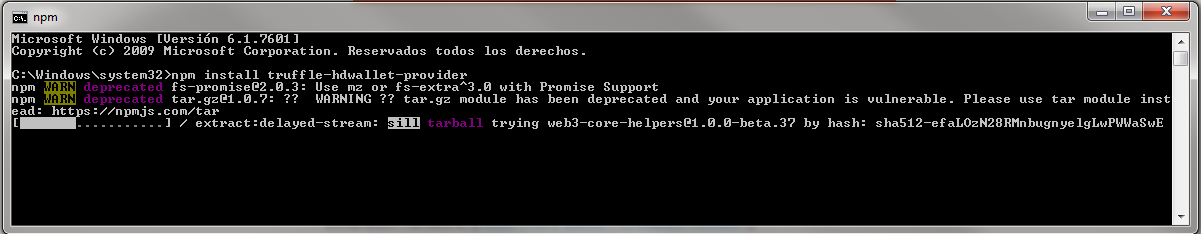


Ilustración 16 Instalacion HDWalletProvider

Para la instalación de ***HDWalletProvider*** es necesario estar ubicado en la raíz del proyecto desde una consola y ejecutar el siguiente comando:

* ***npm install truffle-hdwallet-provider***
  1. Importar truffle en Red privada Ropsten



Ilustración 17 Importar Red Privada Ropsten

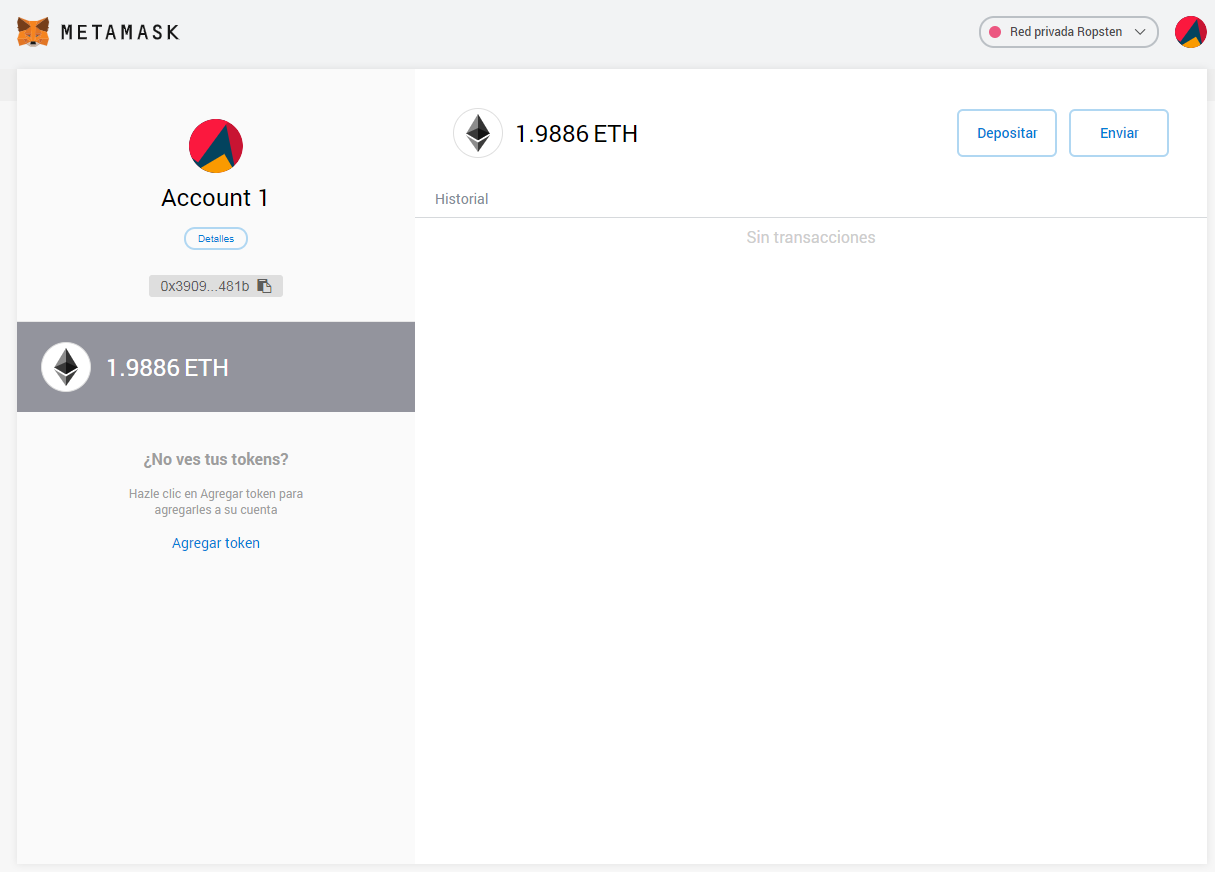


Ilustración 18 Carga de ETH en red pública

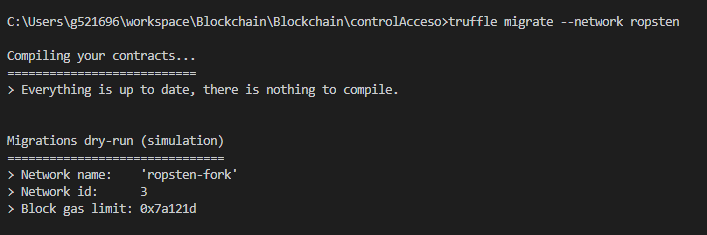


Ilustración 19 Despliegue de contrato en red ropsten

Para realizar el despliegue del contrato en la red ropsten es necesario ubicarse en la raíz del proyecto y ejecutar el siguiente comando:

* ***truffle migrate --network ropsten***

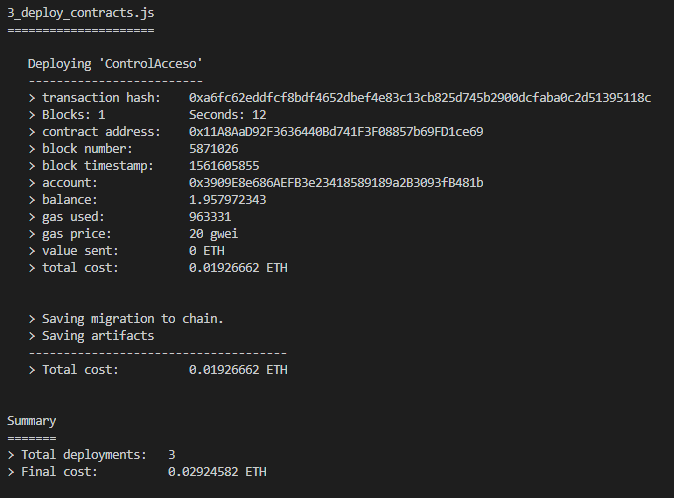


Ilustración 20 Ejecución exitosa de contrato en red Ropsten

Con el contrato desplegado en la red pública ropsten ya se puede ejecutar la aplicación teniendo en cuenta que esta debe cambiar el endpoint a la ruta de infura :

* ***https://ropsten.infura.io/v3/d7e05852912546d3afcd56121efda3d3***

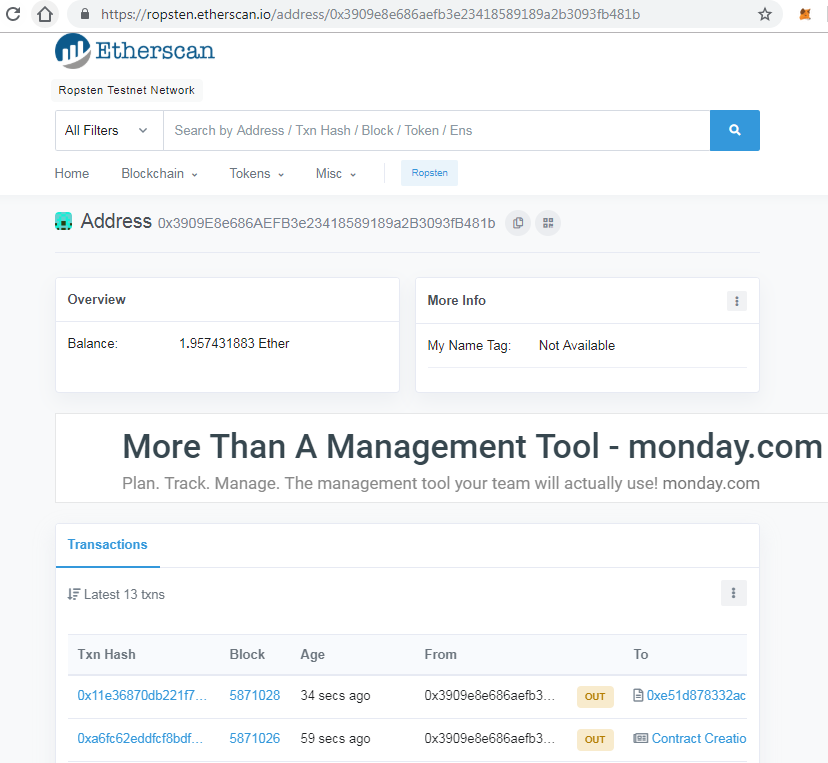


Ilustración 21 Visualización de transacciones sobre red ropsten

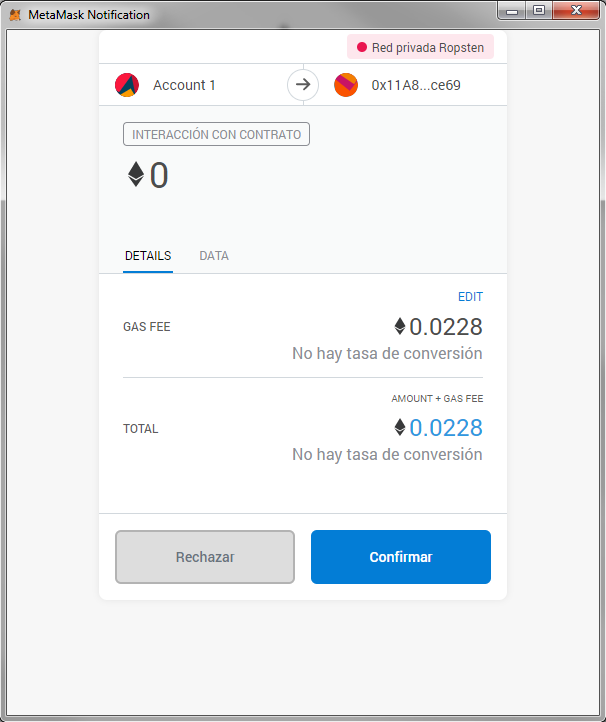


Ilustración 22 Confirmación de transacción sobre ropsten

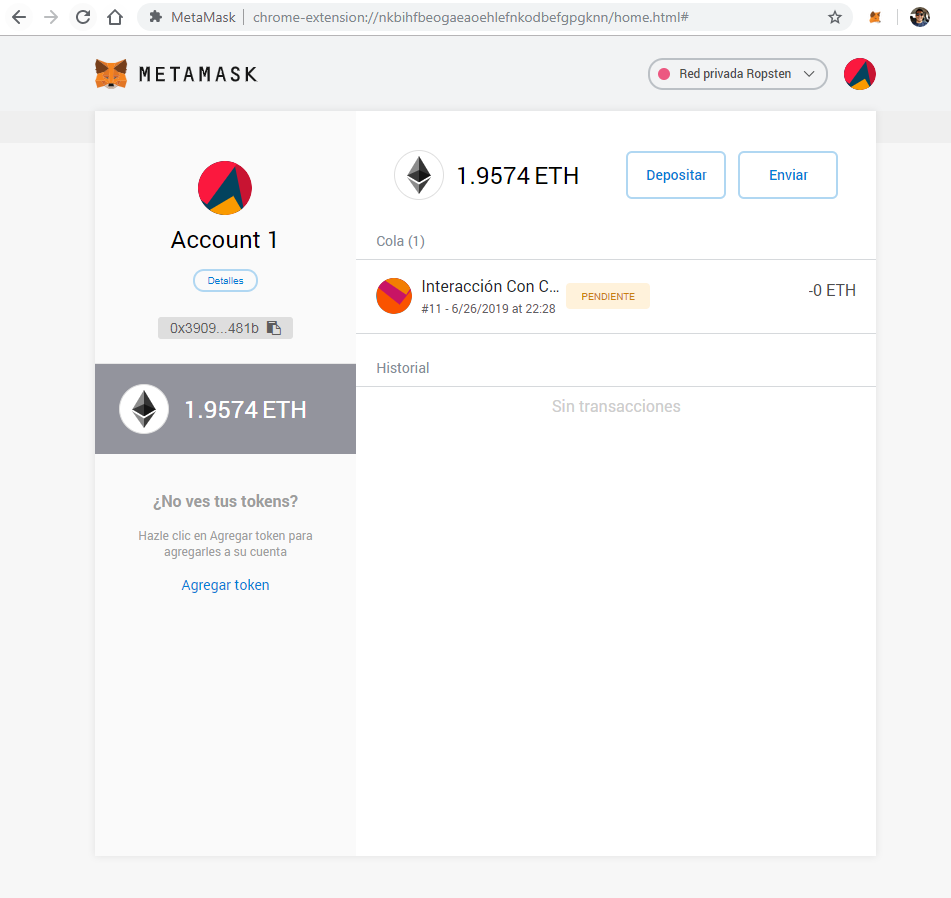


Ilustración 23 Validación de transacción ropsten

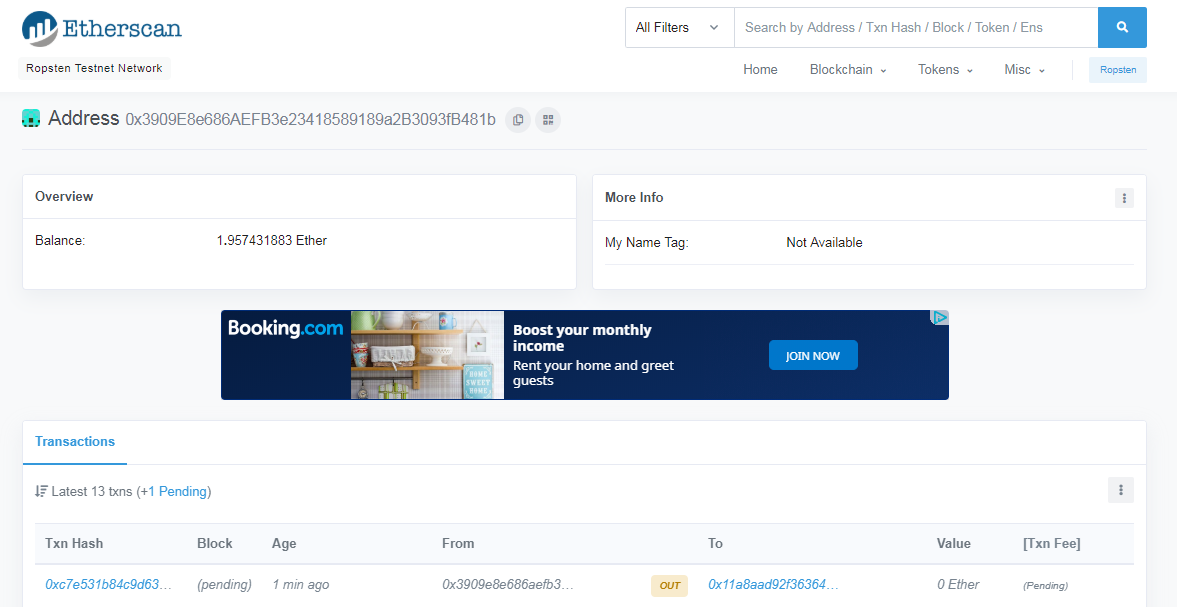


Ilustración 24 Resumen de transacción en red ropsten

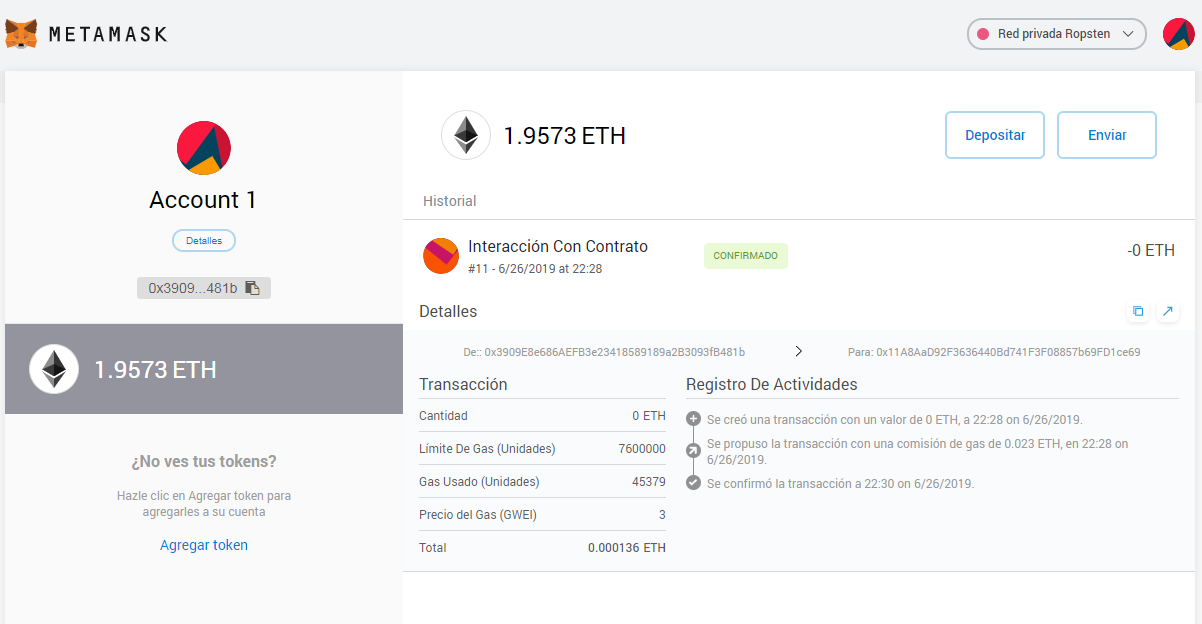


Ilustración 25 Transacción confirmada

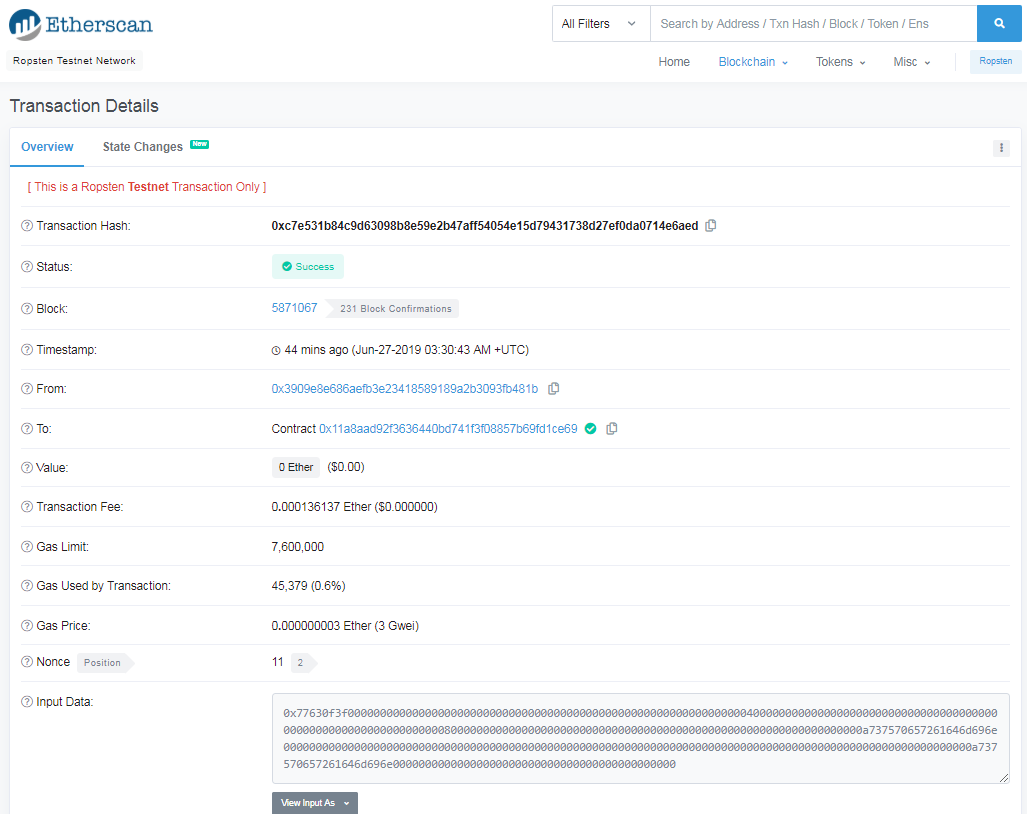


Ilustración 26 Descripción de transacción

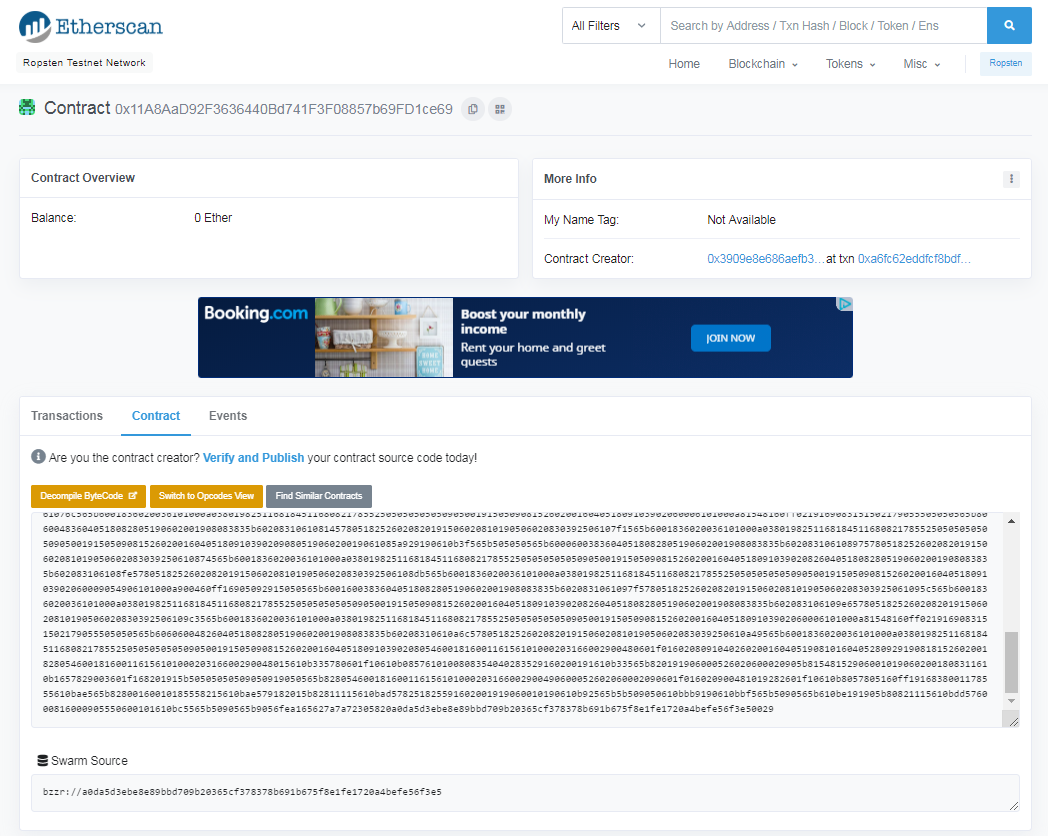


Ilustración 27 Vista de Contrato desplegado en red ropsten

# Código fuente aplicación

El código fuente de la aplicación se encuentra versionada en GITHUBen la siguiente ruta:

* [***https://github.com/yeisontafurc/Blockchain/tree/master/controlAcceso***](https://github.com/yeisontafurc/Blockchain/tree/master/controlAcceso)

Para la ejecución de la aplicación se recomienda utilizar el IDE Visual Studio Code.

1. http://www.isec.com.co/control-de-acceso-que-es-y-para-que-sirve/ [↑](#footnote-ref-1)
2. https://www.xataka.com/especiales/que-es-blockchain-la-explicacion-definitiva-para-la-tecnologia-mas-de-moda [↑](#footnote-ref-2)
3. https://www.trufflesuite.com/docs/truffle/overview [↑](#footnote-ref-3)
4. https://www.trufflesuite.com/docs/ganache/overview [↑](#footnote-ref-4)
5. https://aprendeblockchain.wordpress.com/desarrollo-en-ethereum/desarrollo-con-truffle-i/ [↑](#footnote-ref-5)