

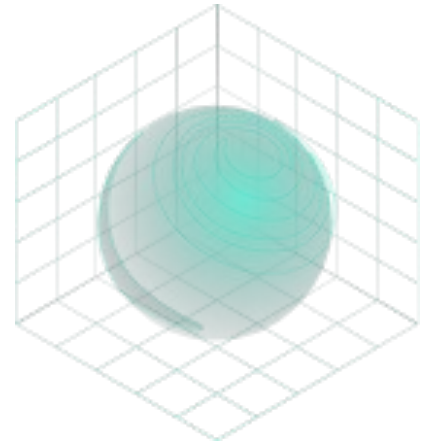
Desarrollo Blockchain Ethereum con Solidity

Módulo 5 - Ejemplo de dApp

Ejemplo de dApp

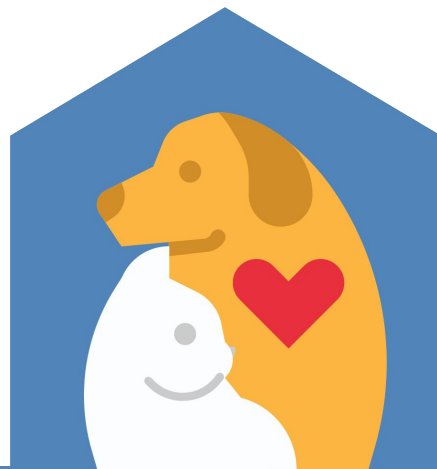
Truffle boxes

- Son soluciones ya desarrolladas disponibles para descargar.
- Soportan diferentes frameworks y lenguajes.
- Podemos ver el listado en <https://www.trufflesuite.com/boxes>



Ejemplo pet-shop

- Utilizaremos este ejemplo que se provee de forma incompleta a propósito.
- Iremos completándolo paso a paso entendiendo cada parte de la construcción de la dApp.
- Para descargar la solución debemos correr en una carpeta vacía el comando `truffle unbox pet-shop`.



Proyecto

Como podemos apreciar, el proyecto tiene la estructura de un proyecto normal de truffle, con el agregado de una carpeta src para el frontend.

1. Lo primero que haremos es ajustar el archivo truffle-config.js para que acepte la última versión del compilador y que además se pueda conectar a ganache.

```
module.exports = {  
  // See <http://truffleframework.com/docs/advanced/configuration>  
  // for more about customizing your Truffle configuration!  
  networks: {  
    ganache: {  
      host: "127.0.0.1",  
      port: 7545,  
      network_id: "*" // Match any network id  
    },  
    develop: {  
      port: 8545  
    }  
  },  
  
  compilers: {  
    solc: {  
      version: "0.8.5"  
    }  
  }  
};
```

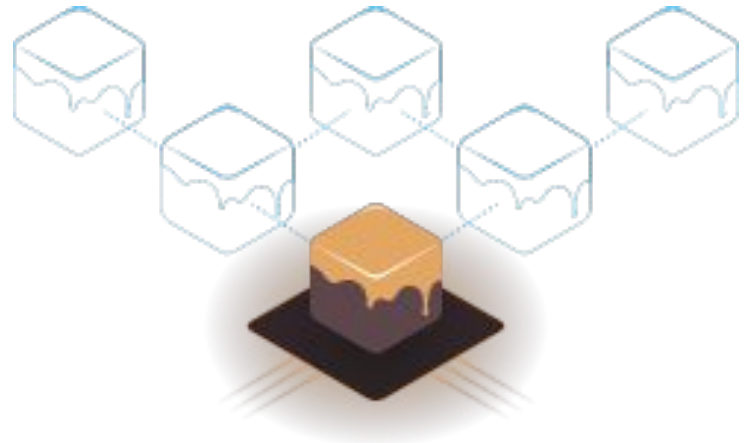
Contrato

2. Escribiremos un contrato nuevo llamado "Adoption.sol" con el siguiente código

```
contract Adoption {  
  
    struct Adopter {  
        string firstname;  
        string lastname;  
        address accountAddress;  
    }  
  
    Adopter[16] public adopters;  
  
    function adopt(uint petId, string memory firstname, string memory lastname) public returns (uint) {  
        require(petId >= 0 && petId <= 15);  
  
        Adopter memory newAdopter = Adopter(firstname, lastname, msg.sender);  
  
        adopters[petId] = newAdopter;  
  
        return petId;  
    }  
  
    function getAdopters() public view returns (Adopter[16] memory) {  
        return adopters;  
    }  
}
```

Migrar a Ganache

3. Ejecutar Ganache y abrir un workspace apuntando al truffle-config.js de este proyecto
4. Abrir la consola de truffle con el comando **truffle console --network ganache**
5. Migrar el proyecto con **migrate --reset**



Conectar Web3 con Metamask

6. Primero abriremos el archivo **app.js** dentro de la carpeta **/src/js/**
7. Buscaremos la función **initWeb3()** y escribiremos el siguiente código para conectar con Metamask.
8. Notar que funciona para navegadores nuevos y viejos.

```
// Modern dapp browsers...
if (window.ethereum) {
  App.web3Provider = window.ethereum;
  try {
    // Request account access
    await window.ethereum.enable();
  } catch (error) {
    // User denied account access...
    console.error("User denied account access")
  }
}
// Legacy dapp browsers...
else if (window.web3) {
  App.web3Provider = window.web3.currentProvider;
}
// If no injected web3 instance is detected, fall back to Ganache
else {
  App.web3Provider = new Web3.providers.HttpProvider('http://localhost:7545');
}
web3 = new Web3(App.web3Provider);
```


Instanciar el contrato

9. En el mismo archivo vamos a guardar una instancia del contrato Adoption para poder llamar a sus funciones desde el frontend.
10. Para esto escribiremos el siguiente código en la función **initContract()**

```
$.getJSON('Adoption.json', function(data) {  
  // Get the necessary contract artifact file and instantiate it with @truffle/contract  
  var AdoptionArtifact = data;  
  App.contracts.Adoption = TruffleContract(AdoptionArtifact);  
  
  // Set the provider for our contract  
  App.contracts.Adoption.setProvider(App.web3Provider);  
  
  // Use our contract to retrieve and mark the adopted pets  
  return App.markAdopted();  
});
```

Cargar las mascotas adoptadas

11. Cargaremos ahora el estado del listado de mascotas, marcando en el sitio aquellas que ya fueron adoptadas
12. Para eso, escribiremos el siguiente código en la función **markAdopted()**

```
var adoptionInstance;

App.contracts.Adoption.deployed().then(function(instance) {
  adoptionInstance = instance;

  return adoptionInstance.getAdopters.call();
}).then(function(adopters) {
  for (i = 0; i < adopters.length; i++) {
    if (adopters[i] !== '0x0000000000000000000000000000000000000000') {
      $('.panel-pet').eq(i).find('button').text('Success').attr('disabled', true);
    }
  }
}).catch(function(err) {
  console.log(err.message);
});
```

Adoptar una mascota

13. Agregaremos el comportamiento al botón de adoptar una mascota, que nos permitirá marcarlo como adoptado por medio de la función `Adopt` del contrato `Adoption`.
14. Para esto agregaremos el siguiente código en la función **`handleAdopt()`**

```
var adoptionInstance;

web3.eth.getAccounts(function(error, accounts) {
  if (error) {
    console.log(error);
  }

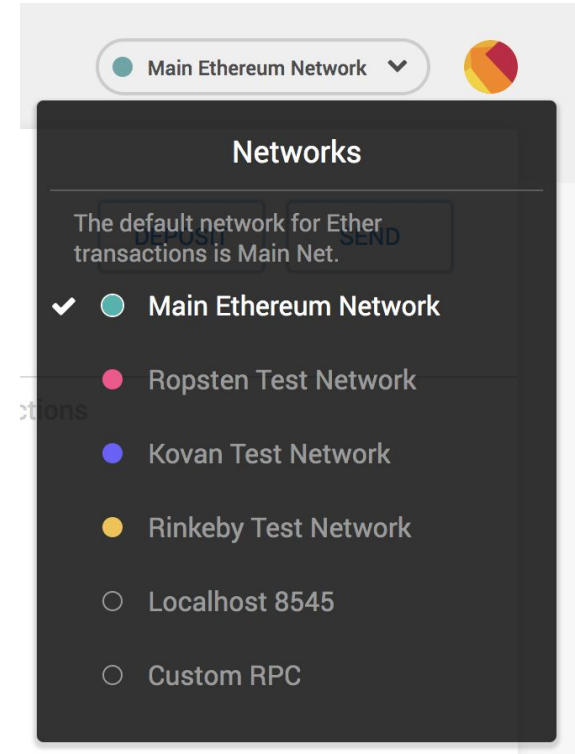
  var account = accounts[0];

  App.contracts.Adoption.deployed().then(function(instance) {
    adoptionInstance = instance;

    // Execute adopt as a transaction by sending account
    return adoptionInstance.adopt(petId, {from: account});
  }).then(function(result) {
    return App.markAdopted();
  }).catch(function(err) {
    console.log(err.message);
  });
});
```

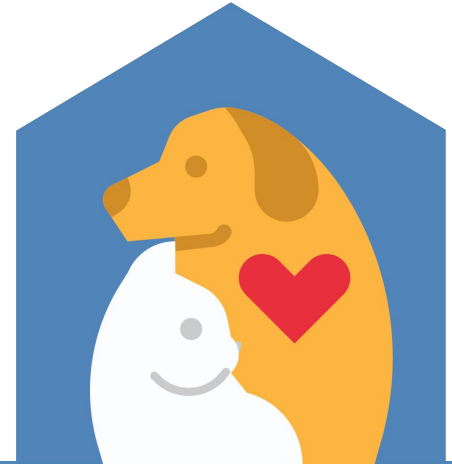
Configurar Metamask

15. Abrir Metamask en el navegador y conectarlo a Ganache por medio de la opción "Custom RPC".
16. Cuando pida los valores, setearle como URL la que nos indique Ganache (en general <http://127.0.0.1:7545>) y como Chain ID el valor 1337.
17. Una vez funcionando, presionar en el avatar circular y seleccionar "Import Account".
18. Donde pide la clave privada, pegarle la clave de alguna de las cuentas de ganache.



Probar la dApp


19. Abrir una terminal y situarse en la carpeta /src.
20. Ejecutar el comando **npm run dev**
21. Abrir el navegador en el sitio que nos indique (en general, localhost:8080).
22. Probar adoptar una mascota y ver que no nos permita volver a adoptarla.
23. Corroborar que al adoptar nos pida confirmar la transacción.



Probar la dApp

Pete's Pet Shop


Frieda



Breed: Scottish Terrier
Age: 3
Location: Lisso, Alabama

Adopt


Gina



Breed: Scottish Terrier
Age: 3
Location: Tooleville, West Virginia

Adopt


Collins



Breed: French Bulldog
Age: 2
Location: Freeburn, Idaho

Adopt


Melissa



Breed: Boxer
Age: 2
Location: Camas, Pennsylvania

Adopt


Jeanine



Breed: French Bulldog
Age: 2
Location: Gerber, South Dakota

Adopt


Elvia



Breed: French Bulldog
Age: 3
Location: Innsbrook, Illinois

Adopt


Latisha



Breed: Golden Retriever
Age: 3
Location: Soudan, Louisiana

Adopt

Coleman



Breed: Golden Retriever
Age: 3
Location: Jacksonwald, Palau

Adopt

Nichole

Fran

Leonor

Dean

http://127.0.0.1:7545

Account 1 → 0x25B9...Fa5A

CONTRACT INTERACTION

0

DETAILS DATA

EDIT

GAS FEE 0.000379
No Conversion Rate Available

AMOUNT + GAS FEE

TOTAL 0.000379
No Conversion Rate Available

REJECT CONFIRM

¡Muchas gracias!

¡Sigamos trabajando!