

Laboratorios

Módulo 06

Ejercicio 1: Implementar un token ERC-20

1. Crear un proyecto nuevo de Truffle con el comando **truffle init**.
2. Instalar Open Zeppelin en el proyecto con el comando **npm install @openzeppelin/contracts**
3. Opcional: configurar **truffle-config.js** para que se conecte con Ganache.
4. Crear un contrato MiToken.sol
5. Importar el contrato ERC20 de OpenZeppelin (ver código abajo).
6. Heredar el contrato MiToken de ERC20.

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;

import "../node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol";

contract MiToken is ERC20 {

}
```

Ejercicio 2: Expandir la funcionalidad del Token

1. Agregar un atributo de tipo string para escribir una descripción del token.
2. Agregar una función para ver el contenido de esa descripción.
3. Redefinir el constructor para que acepte por parámetro la descripción y así asignársela a la variable.
4. Agregar la función **_mint** en el constructor para que el sender tenga el supply inicial en su balance.
5. Deployarlo y corroborar que el balance del sender sea 1000.

```
contract MiToken is ERC20 {  
    string descripcion;  
  
    constructor(string memory name, string memory symbol, string memory _descripcion) is ERC20(name, symbol) {  
        descripcion = _descripcion;  
        _mint(msg.sender, 1000)  
    }  
}
```

¡Muchas gracias!

¡Sigamos trabajando!