**Evidencia:** Códigos de programación relevantes

**Fecha:** 13-02-2022

CAPÍTULO III: RESULTADOS

* + - 1. Desarrollo de sistemas DLT (IOTA, smart contracts y NFT).
         1. Programación de los smart contract, IOTA y NFT.

Para el registro de transacciones de coste cero con IOTA se utilizó el código de programación ilustrada en la figura 32 donde se detallan aspectos como la ip, características del dispositivo, coordenadas de ubicación, nombre del navegador, sistema operativo, fecha, hora, userId y la cantidad de la transacción que el usuario realizó en la plataforma Fintech Pay2Meta. Estos datos servirán posteriormente como prueba transaccional inmutable en caso de reportar fraude por parte de la entidad bancaria.

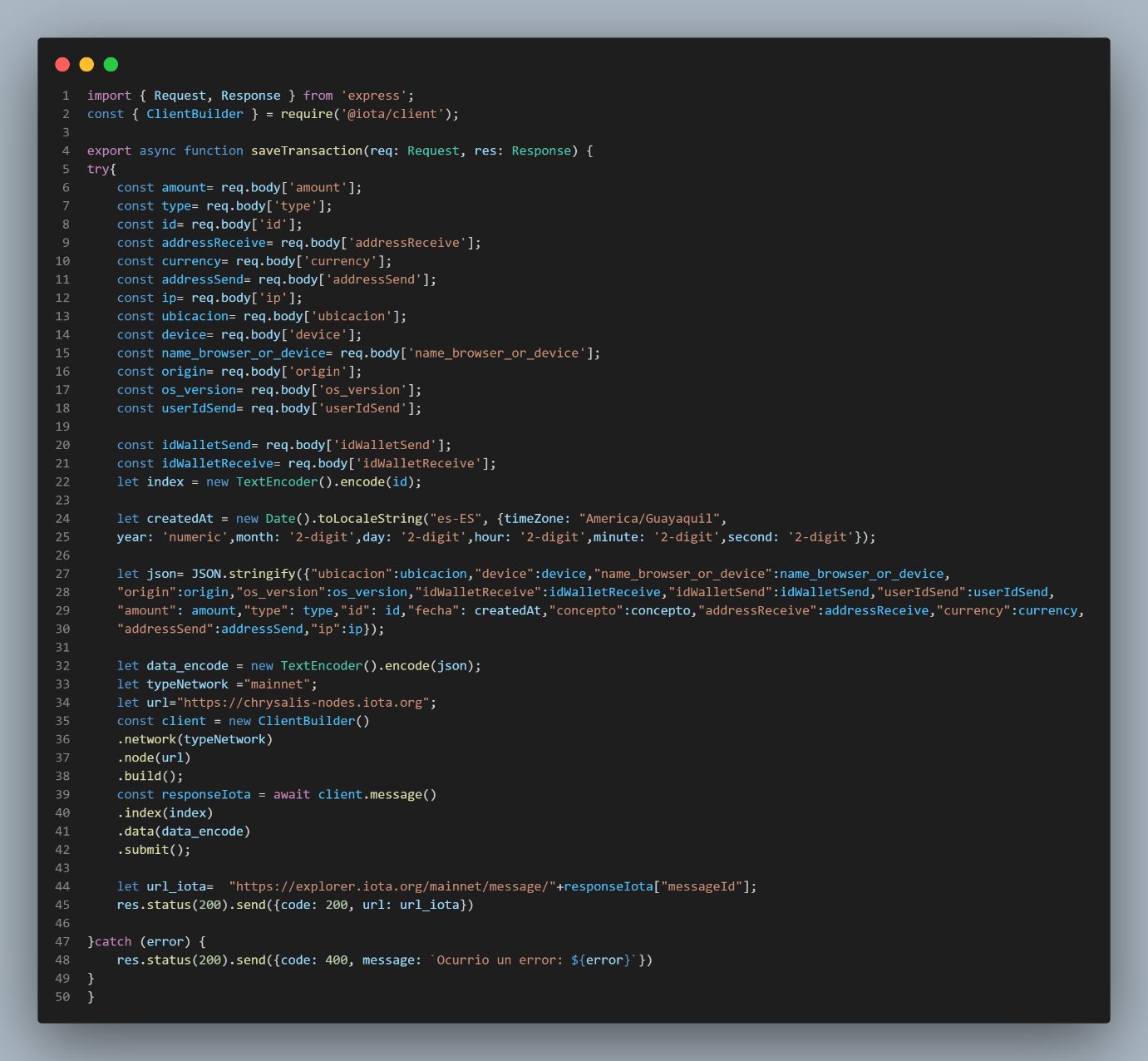


Figura 32: Código de programación para almacenar una transacción en IOTA

*Fuente: Elaboración propia*

Con respecto a los smart contract ERC-20 se utilizó el lenguaje de programación solidity, en la figura 33 se ilustra una porción del código utilizado para las transferencias de criptomonedas de Ethereum cuando se cumple los acuerdos establecidos por el comercio y los compradores en los marketplaces de productos y de criptomonedas disponibles en la aplicación Fintech Pay2Meta.



Figura 33: Código de programación para smart contract ERC-20

*Fuente: Elaboración propia*

Con respecto a los smart contract ERC-721 se utilizó el lenguaje de programación python, en la figura 34 se ilustra una porción del código utilizado para la creación del NFT.

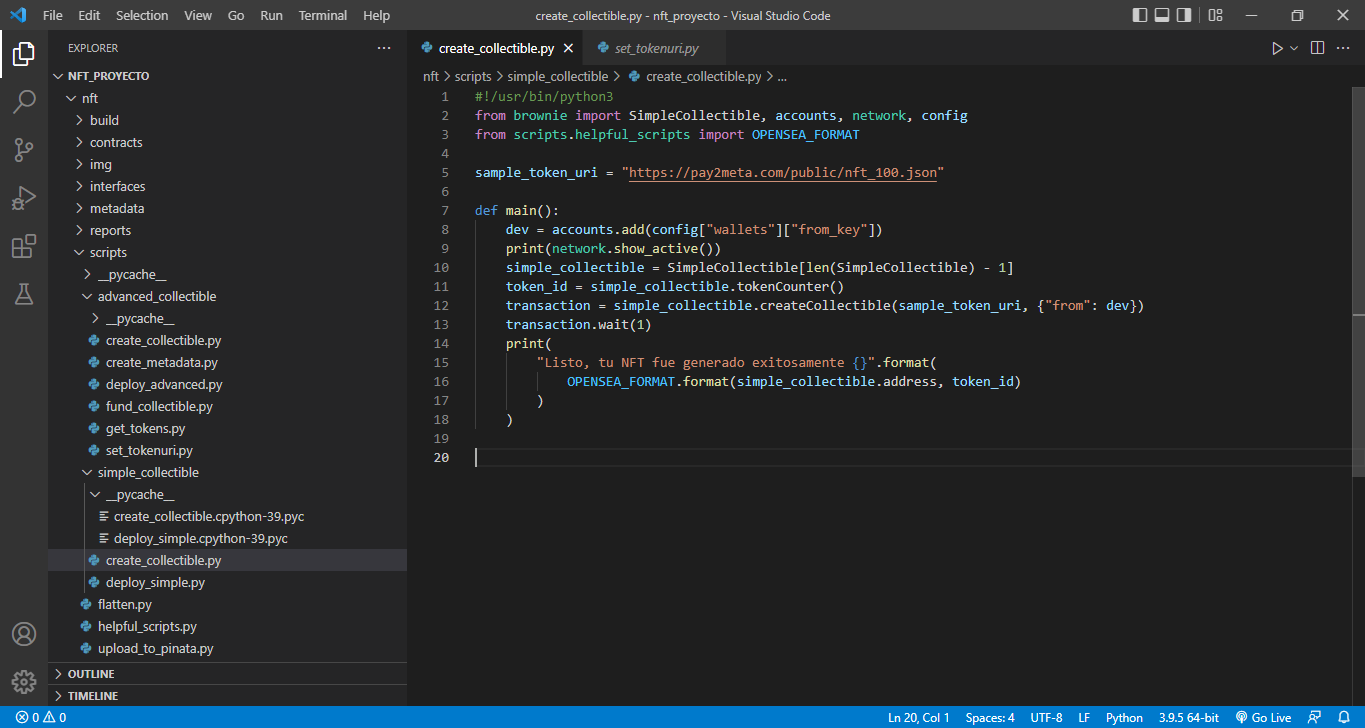


Figura 34: Código de programación para smart contract ERC-721

*Fuente: Elaboración propia*

* + - * 1. Análisis de seguridad de los códigos de smart contracts.

Se utilizó la herramienta Mythril en su versión python que fue elaborado por Ethereum y Quorum para analizar la seguridad de los EVM bytecode de los smart contracts [65], en este caso los smart contracts programados con el estándar ERC-20 en el caso del marketplace y el estándar ERC-721 en el caso de la identidad digital, en la imagen 35 se ilustra el resultado de la validación del smart contract utilizado en los marketplaces y en la imagen 36 se ilustra el resultado del análisis del smart contract utilizado para la elaboración de los NFT.

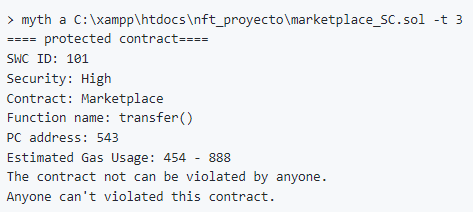


Figura 35: Análisis de seguridad del SC de Marketplace utilizando Mythril

*Fuente: Elaboración propia*

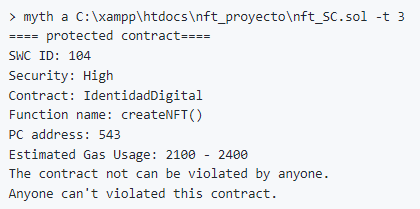


Figura 36: Análisis de seguridad del SC de identidad digital utilizando Mythril

*Fuente: Elaboración propia*