

Smart Contract Developer Challenge

En nuestro proyecto estrella vamos a entregar NFTs que representan las monedas y billetes de un país con distintas denominaciones.

- 1 peso
- 5 pesos
- 20 pesos
- 10 pesos
- 50 pesos
- 100 pesos

Necesitamos entregar el mínimo de cada NFTs por cada transacción. Tu misión es calcular la función que convierte un número en una lista de NFTs mínimos para entregar.

Ejemplos:

65 = 50,10,5

98 = 50,20,20,5,1,1,1

341 = 100,100,100,20,20,1

Para cada denominación existe un stock limitado de NFTs, por lo tanto no podemos dar los que no existen.

Stock:

- 1 = 50 unidades
- 5 = 5 unidades
- 20 = 2 unidades
- 50 = 2 unidades
- 100 = 20 unidades

Debes programar un Smart Contract usando el lenguaje Solidity, que tenga una query donde se le entregue como input un número entero y retorne los valores que corresponden al desglose según la denominación. Este Smart Contract debe llevar la cuenta de cuantos NFTs quedan de cada uno (si no quedan de 20, debe entregar ahora con los de 5 o de 1 que queden). Este contrato debe ser desplegado en testnet para que podamos hacer queries y te vamos a pedir que nos expongas la solución en una entrevista.

Debes incluir las siguientes queries:

- ConvertDenom (int) //retorna la conversión de monedas
- ChangeStock([int, int, int, int, int]) //actualiza el stock disponible de cada denominación. Solo debe ser accionable desde la cuenta admin.

Debes subir el progreso de tu proyecto a una cuenta personal de GitHub como proyecto público para que podamos revisar.

Qué se evaluará:

- La prolijidad del repositorio y la calidad del código.
- Que la solución funcione sin errores y tenga las validaciones para evitarlos.

Ganas (muchos) puntos extra si:

- Incluye un frontend simple donde podamos hacer las llamadas con una wallet Metamask a través de una UI
- Incluye pruebas unitarias