







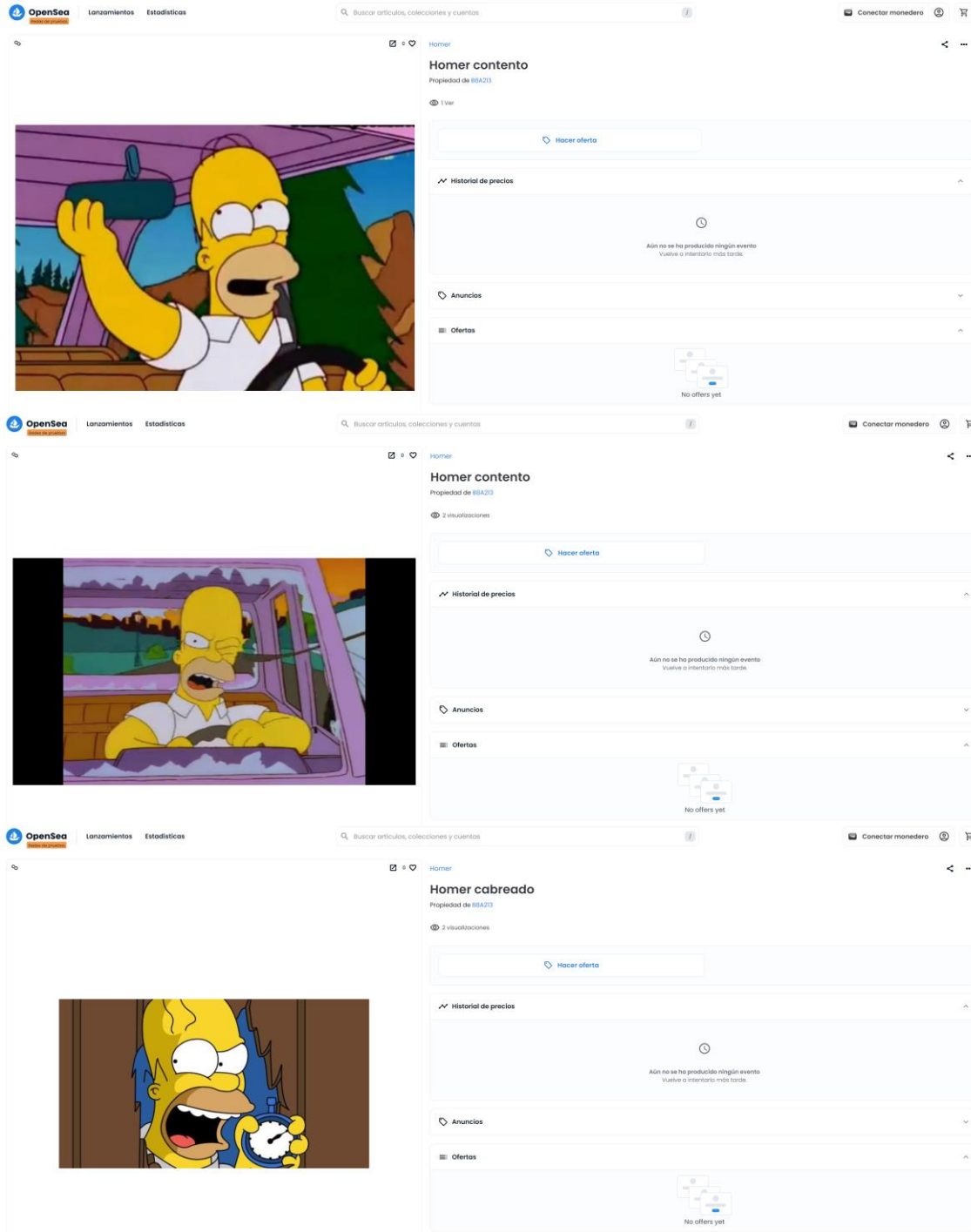
Contrato de primer minteo (ausente de automatización)

```

1  // SPDX-License-Identifier: MIT
2
3  //Creado por Jonatan Gomez Garcia gracias a las clases de Bitlab
4  pragma solidity ^0.8.9;
5
6  import "@openzeppelin/contracts@4.8.2/token/ERC721/ERC721.sol";
7  import "@openzeppelin/contracts@4.8.2/token/ERC721/extensions/ERC721URIStorage.sol";
8  import "@openzeppelin/contracts@4.8.2/access/Ownable.sol";
9  import "@openzeppelin/contracts@4.8.2/utils/Counters.sol";
10
11 contract Homer is ERC721, ERC721URIStorage, Ownable {
12     using Counters for Counters.Counter;
13
14     string[] IpfsUri = [
15         "https://gateway.pinata.cloud/ipfs/QmavFFBcRNvpUHHj9Bbv7ZqhF3zibzAsKUJNJs7J1YAZT3/homer_0.json",
16         "https://gateway.pinata.cloud/ipfs/QmavFFBcRNvpUHHj9Bbv7ZqhF3zibzAsKUJNJs7J1YAZT3/homer_1.json",
17         "https://gateway.pinata.cloud/ipfs/QmavFFBcRNvpUHHj9Bbv7ZqhF3zibzAsKUJNJs7J1YAZT3/homer_2.json"
18     ];
19
20     Counters.Counter private _tokenIdCounter;
21
22     constructor() ERC721("Homer", "HOM") {}  infinite gas 2558200 gas
23
24     function safeMint(address to) public {  infinite gas
25         uint256 tokenId = _tokenIdCounter.current();
26         _tokenIdCounter.increment();
27         _safeMint(to, tokenId);
28         _setTokenURI(tokenId, IpfsUri[0]);
29     }
30
31     function pymyRide(uint256 _tokenId) public {  infinite gas
32         require(homerLevel(_tokenId) < 2, "Homer totalmente cabreado!");
33
34         // obtiene el valor actual del auto y le suma 1
35         uint256 newVal = homerLevel(_tokenId) + 1;
36         // store the new URI
37         string memory newUri = IpfsUri[newVal];
38         //Update the URI
39         _setTokenURI(_tokenId, newUri);
40     }
41
42     // helper functions
43     //
44     function homerLevel(uint256 _tokenId) public view returns(uint256){  infinite gas
45         string memory _uri = tokenURI(_tokenId);
46
47         uint result;
48         //Guardamos el factor comparador para una mejor eficiencia en cuanto a consumo de gas
49         bytes32 toFind = keccak256(abi.encodePacked(_uri));
50
51         for (uint256 index = 0; index < IpfsUri.length; index++) {
52             if(keccak256(abi.encodePacked(IpfsUri[index])) == toFind)
53                 result = index;
54         }
55         return result;
56     }
57
58     // The following functions are overrides required by Solidity.
59
60     function _burn(uint256 tokenId) internal override(ERC721, ERC721URIStorage) {  infinite gas
61         super._burn(tokenId);
62     }
63
64     function tokenURI(uint256 tokenId)  infinite gas
65         public
66         view
67         override(ERC721, ERC721URIStorage)
68         returns (string memory)
69     {
70         return super.tokenURI(tokenId);
71     }
72 }
73

```

Como se observa en el código anterior he aplicado una pequeña mejora en términos de rendimiento de gas guardando el factor comparador en forma de variable en vez de calcularlo por cada vuelta de bucle, para este caso es casi imperceptible, ya que solo trabajamos con tres niveles o vueltas dentro del bucle. Sería interesante qué otras opciones de optimización son posibles.



Contrato con automatización

```

1 // SPDX-License-Identifier: MIT
2
3 //Creado por Jonatan Gomez Garcia gracias a las clases de Bitlab
4 pragma solidity ^0.8.9;
5
6 import "@openzeppelin/contracts@4.8.2/token/ERC721/ERC721.sol";
7 import "@openzeppelin/contracts@4.8.2/token/ERC721/extensions/ERC721URIStorage.sol";
8 import "@openzeppelin/contracts@4.8.2/access/Ownable.sol";
9 import "@openzeppelin/contracts@4.8.2/utils/Counters.sol";
10 import "@chainlink/contracts/src/v0.8/AutomationCompatible.sol";
11
12 contract Homer is ERC721, ERC721URIStorage, Ownable, AutomationCompatibleInterface {
13     using Counters for Counters.Counter;
14
15     string[] IpfsUri = [
16         "https://gateway.pinata.cloud/ipfs/QmavFFBcRNvpUHHj9Bbv7Zqhf3zibzAskUJN3s7J1YAZT3/homer_0.json",
17         "https://gateway.pinata.cloud/ipfs/QmavFFBcRNvpUHHj9Bbv7Zqhf3zibzAskUJN3s7J1YAZT3/homer_1.json",
18         "https://gateway.pinata.cloud/ipfs/QmavFFBcRNvpUHHj9Bbv7Zqhf3zibzAskUJN3s7J1YAZT3/homer_2.json"
19     ];
20
21     Counters.Counter private _tokenIdCounter;
22
23     uint interval;
24     uint lastTimeStamp;
25
26     constructor(uint _interval) ERC721("CardNFT", "dNFT") {
27         interval = _interval;
28         lastTimeStamp = block.timestamp;
29     }
30
31
32     function safeMint(address to) public {
33         uint256 tokenId = _tokenIdCounter.current();
34         _tokenIdCounter.increment();
35         _safeMint(to, tokenId);
36         _setTokenURI(tokenId, IpfsUri[0]);
37     }
38
39     function pympMyRide(uint256 _tokenId) public {
40         require(homerLevel(_tokenId) < 2, "Homer totalmente cabreado!");
41
42         // obtiene el valor actual del auto y le suma 1
43         uint256 newVal = homerLevel(_tokenId) + 1;
44         // store the new URI
45         string memory newUri = IpfsUri[newVal];
46         //Update the URI
47         _setTokenURI(_tokenId, newUri);
48     }
49
50     // Helper functions
51     //
52     function homerLevel(uint256 _tokenId) public view returns(uint256){
53         string memory _uri = tokenURI(_tokenId);
54
55         uint result;
56         //Guardamos el factor comparador para una mejor eficiencia en cuanto a consumo de gas
57         bytes32 toFind = keccak256(abi.encodePacked(_uri));
58
59         for (uint256 index = 0; index < IpfsUri.length; index++) {
60             if(keccak256(abi.encodePacked(IpfsUri[index])) == toFind)
61                 result = index;
62         }
63         return result;
64     }
65
66     function checkUpkeep(bytes calldata /* checkData */) external view override returns (bool upkeepNeeded, bytes memory /* performData */) {
67         upkeepNeeded = (block.timestamp - lastTimeStamp) > interval;
68     }
69
70     function performUpkeep(bytes calldata /* performData */) external override {
71         if ((block.timestamp - lastTimeStamp) > interval) {
72             lastTimeStamp = block.timestamp;
73
74             uint counter = _tokenIdCounter.current();
75
76             for (uint256 index = 0; index < counter; index++) {
77                 if(homerLevel(index) < 2){
78                     pympMyRide(index);
79                     break;
80                 }
81             }
82         }
83     }
84
85     // The following functions are overrides required by Solidity.
86
87     function _burn(uint256 tokenId) internal override(ERC721, ERC721URIStorage) {
88         super._burn(tokenId);
89     }
90
91     function tokenURI(uint256 tokenId) public
92         view
93         override(ERC721, ERC721URIStorage)
94         returns (string memory)
95     {
96         return super.tokenURI(tokenId);
97     }
98 }
99
100

```

[Home](#) /

Register new Upkeep

Welcome to Chainlink Automation - fully automate your contract in two simple steps.

Trigger

Custom logic



Target contract address



0x5198BbDe02A3D967040656C2488eD0D4b036cC2C



[Edit](#)

Upkeep details

Upkeep name

Homer Dinamic NFT

Provide a name for your Upkeep to easily manage it in the Automation UI.

Admin Address

0xb8a213e553519fdb4312c353b32eb51f51ad79c

The address for the administrator of this Upkeep.

Gas limit

1500000

Amount of gas to provide the target contract when performing Upkeep. This will impact minimum balance requirements, and should be approximately the maximum amount of gas the transaction might use.

Starting balance (LINK)

5

Deposit LINK to your Upkeep. Select an amount that will satisfy multiple performances to start, then fund the Upkeep directly once it's operational.

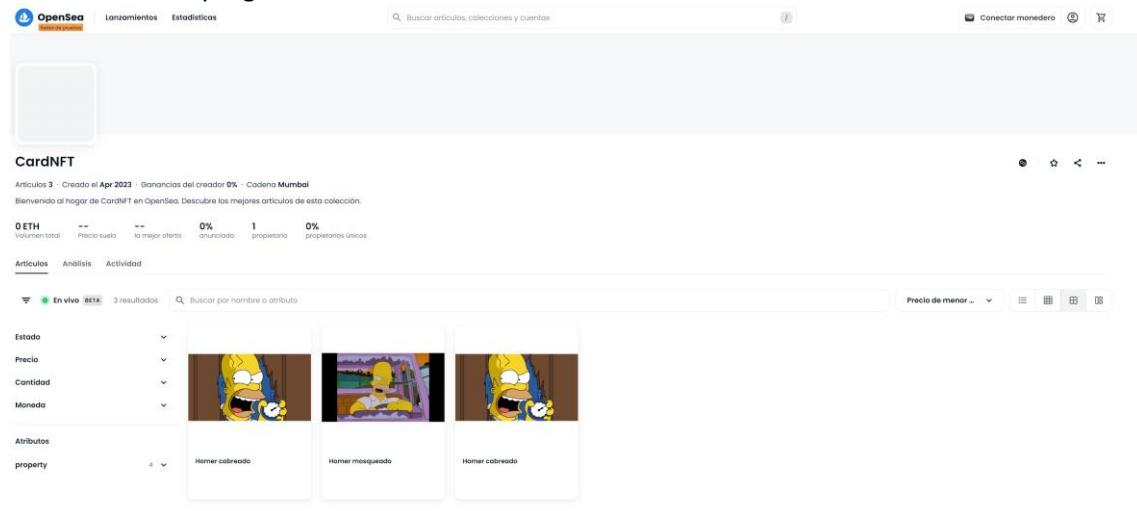
Check data (Hexadecimal) *Optional*

Pass static data into your checkUpkeep function. This will be converted to bytes. See [docs](#) for details.



Need LINK for testing? Visit the [Chainlink Polygon Mumbai Faucet](#) to receive testnet LINK.

Podemos ver el progreso



OpenSea

CardNFT

Artículo 3 · Creado el Apr 2023 · Ganancias del creador 0% · Codena Mumbai

Bienvenido al hogar de CardNFT en OpenSea. Descubre los mejores artículos de esta colección.

0 ETH Volumen total · -- Precio suelo · -- la mejor oferta · 0% anunciado · 1 propietario · 0% propietarios únicos

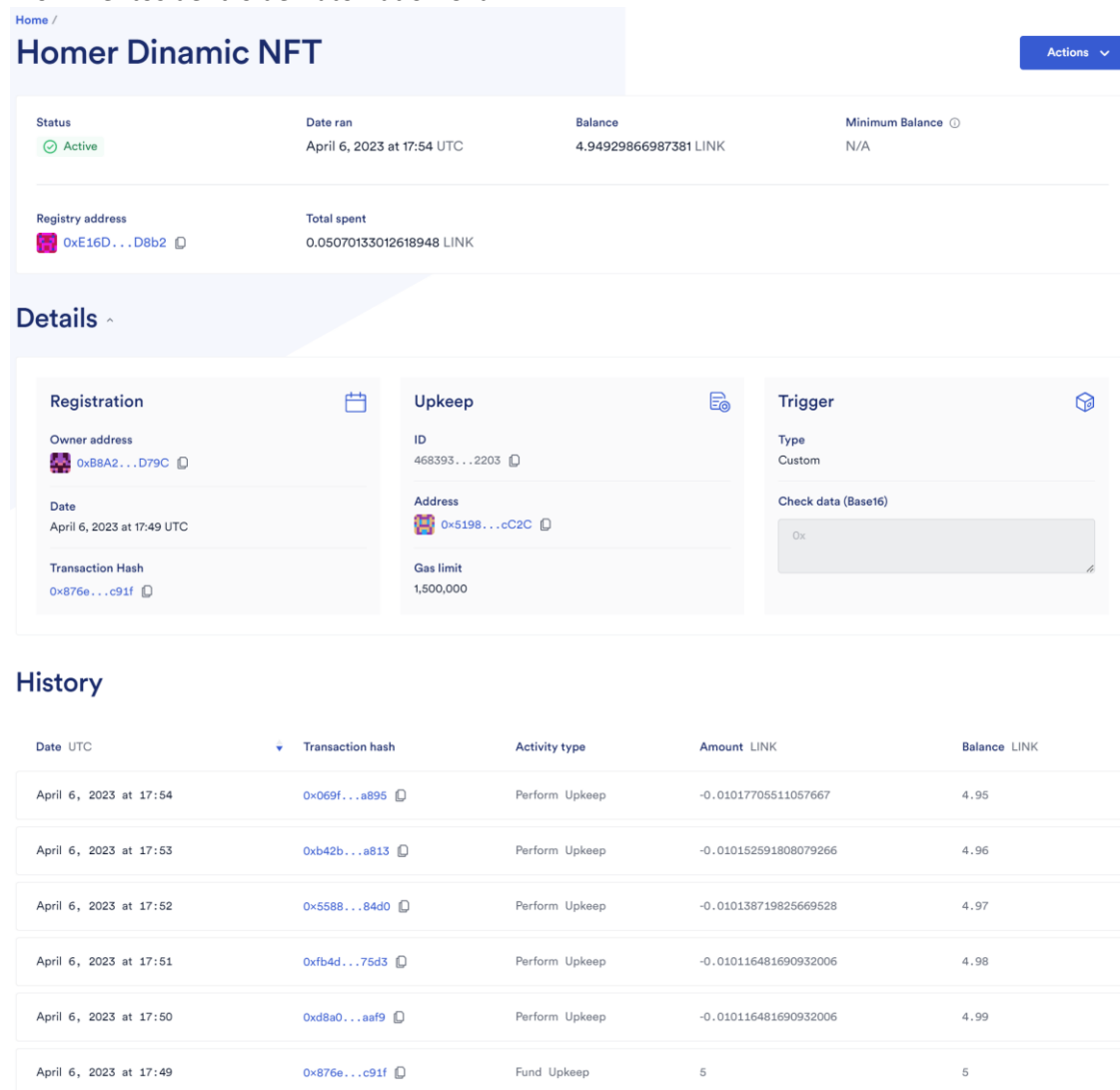
Artículos · Análisis · Actividad

En vivo · 3 resultados · Buscar por nombre o atributo · Precio de menor ...

Estado · Precio · Cantidad · Moneda · Atributos

property · Homer catbreed · Homer mosquero · Homer catbreed

Finalmente, todos los NFT en su último nivel, tenemos 3 NFT, evolucionan 2 veces: resultado 6 movimientos dentro de Automation Chainlink



Home /

Homer Dinamic NFT

Actions

Status: Active

Date ran: April 6, 2023 at 17:54 UTC

Balance: 4.94929866987381 LINK

Minimum Balance: N/A

Registry address: 0xE16D...D8b2

Total spent: 0.05070133012618948 LINK

Details

Registration

Owner address: 0xB8A2...D79C

Date: April 6, 2023 at 17:49 UTC

Transaction Hash: 0x876e...c91f

Upkeep

ID: 468393...2203

Address: 0x5198...c2C2

Gas limit: 1,500,000

Trigger

Type: Custom

Check data (Base16): 0x

History

Date UTC	Transaction hash	Activity type	Amount LINK	Balance LINK
April 6, 2023 at 17:54	0x069f...a895	Perform Upkeep	-0.01017705511057667	4.95
April 6, 2023 at 17:53	0xb42b...a813	Perform Upkeep	-0.010152591808079266	4.96
April 6, 2023 at 17:52	0x5588...84d0	Perform Upkeep	-0.010138719825669528	4.97
April 6, 2023 at 17:51	0xfb4d...75d3	Perform Upkeep	-0.010116481690932006	4.98
April 6, 2023 at 17:50	0xd8a0...aa19	Perform Upkeep	-0.010116481690932006	4.99
April 6, 2023 at 17:49	0x876e...c91f	Fund Upkeep	5	5