

# Funciones del Smart-Contract

## Parte de lectura del contrato:

1. **balanceOf(address owner) → uint256 balance:** Devuelve el número de NFT en la cuenta del propietario.
2. **BaseExtension:** .json (formato de la metadata) (string)
3. **Cost:** costo en Wei por mint un token
4. **getApproved(uint256 tokenId) → address:** Obtiene la dirección aprobada para un identificador de token, o cero si no se ha establecido ninguna dirección. Se revierte si el identificador de token no existe.
5. **isApprovedForAll(address owner, address operator) → bool:** Indica si un operador está aprobado por un propietario determinado.
6. **maxMintAmount:** indica la cantidad que se pueden mintear por sesión.
7. **maxSupply:** indica la cantidad máxima de NFTs que se pueden mintear.
8. **name:** Nombre del token.
9. **notRevealUri:** muestra el baseUrl de la imagen almacenada que se usa cuando revealed (bool: false).
10. **owner:** address del que lanzo el contrato.
11. **ownerOf(uint256 tokenId) → address owner:** Devuelve el propietario del NFT especificado por tokenId.
12. **Paused:** lectura del estado (bool: false or true).
13. **Revelated:** lectura del estado (bool: false or true).
14. **supportsInterface(bytes4 interfacedId) → bool:** Complejidad de tiempo  $O(1)$ , garantizado para usar siempre menos de 30 000 gases.
15. **Symbol:** abreviación del nombre.
16. **tokenByIndex(uint256 index) → uint256:** Devuelve una identificación de token en un dado index de todos los tokens almacenados por el contrato. Use junto con [totalSupply](#) para enumerar todos los tokens.
17. **tokenOfOwnerByIndex(address owner, uint256 index) → uint256 tokenId:** Devuelve un ID de token propiedad de owner una determinada index de su lista de tokens.
18. **tokenURI(uint256 tokenId) → string:** Devuelve el identificador uniforme de recursos (URI) para el tokenId token.
19. **totalSupply() → uint256:** devuelve la cantidad de token minteados, es decir, la cantidad de tokens almacenados por el contrato.
20. **walletOfOwner:** indica que tokens tiene un propietario.

**Parte de escritura del contrato:** Funciones que pueden modificar cualquiera (Publico), funciones que solo el propietario puede modificar (Interno).

1. **approve(address to, uint256 tokenId):** Otorga permiso para transferir el token tokenId a otra cuenta. La aprobación se borra cuando se transfiere el token. Solo se puede aprobar una sola cuenta a la vez, por lo que aprobar la dirección cero borra las aprobaciones anteriores. Requisitos: 1) La persona que llama debe poseer el token o ser un operador aprobado. 2) tokenId debe existir. Emite un evento de aprobación. (Publico)
2. **mint:** almacena un token en el contrato obtenido del Uri. Tiene dos funciones: payableAmount (costo del minteo) y \_mintAmount (uint256) (cantidad de token que se van a mintear). (Publico)
3. **Pause (bool):** activa o desactiva las funciones del contrato. (Interno)
4. **renounceOwnership:** se renuncia a la propiedad del contrato. (Interno)
5. **Reveal (bool):** revela la imagen del NFTs o viceversa. (Interno)
6. **safeTransferFrom(address from, address to, uint256 tokenId):** Transfiere de forma segura la propiedad de una ID de token determinada a otra dirección. (Publico)
7. **safeTransferFrom(address from, address to, uint256 tokenId, bytes \_data):** Transfiere de forma segura la propiedad de una ID de token determinada a otra dirección. Si la dirección de destino es un contrato, debe implementar onERC721Received, lo que requiere una transferencia segura, y devolver el valor mágico bytes4(keccak256("onERC721Received(address,address,uint256,bytes)")); de lo contrario, la transferencia se revierte. Requiere que msg.sender sea el propietario, aprobado u operador. (Interno)
8. **setApprovalForAll(address to, bool approved):** Establece o deshabilita la aprobación de un operador determinado Un operador puede transferir todos los tokens del remitente en su nombre. (Publico)
9. **setBaseExtension (string):** se usa para cambiar el formato (.json). (Interno)
10. **setBaseURI(string baseURI):** Función interna para establecer el URI base para todas las ID de token. Se agrega automáticamente como prefijo al valor devuelto en tokenURI. (Interno)
11. **setCost:** modificar el costo del mint de un token en Wei. (Interno)
12. **setNotRevealedUri (string):** se usa para cambiar el Uri que aparece como imagen cuando la función "Reveal" es falsa. (Interno)
13. **setMaxMintAmount:** se usa para modificar la cantidad de tokens que se pueden mintear por sesión. (Interno)
14. **trasferFrom (address from, address to, unit256 tokenId):** Función interna para transferir la propiedad de una identificación de token determinada a otra dirección. A diferencia de [transferFrom](#), esto no impone restricciones a msg.sender. (Interno)

**15. transferOwnership:** transferir a otra dirección la propiedad del contrato.  
(interno)

**16. withdraw:** función para recaudar todo el dinero almacenado en el contrato. (Interno)