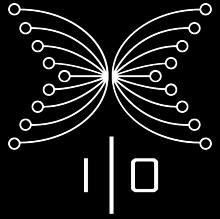


INPUT | OUTPUT

# Minting Policy



1

# MINTING POLICIES

## Minting Policy : Con Aiken

---

A diferencia de los Spend Validators en Aiken, los Minting scripts no usan el Datum como parámetro:

```
validator free_tokens {
    spend(_datum, _redeemer, _utxo, _tx) {
        todo @"spend"
    }

    mint(_redeemer, _policy_id, _tx) {
        todo @"mint"
    }
    // ...
}
```

## Minting Policy : Con Aiken

---

A diferencia de los Spend Validators en Aiken, los Minting scripts no usan el Datum como parámetro:

```
validator free_tokens {
    spend(_datum, _redeemer, _utxo, _tx) {
        todo @"spend"
    }

    mint(_redeemer, _policy_id, _tx) {
        todo @"mint"
    }
    // ...
}
```

El Spend validator necesita:  
**Datum, Redeemer, Output Reference y Transaction**

## Minting Policy : Con Aiken

---

A diferencia de los Spend Validators en Aiken, los Minting scripts no usan el Datum como parámetro:

```
validator free_tokens {
    spend(_datum, _redeemer, _utxo, _tx) {
        todo @"spend"
    }

    mint(_redeemer, _policy_id, _tx) {
        todo @"mint"
    }
    // ...
}
```

El Mint validator solo necesita:  
**Redeemer, Policy id y Transaction**

## Minting Policy : partes de un token

---

Un token está compuesto por dos partes: **Policy Id + Token Name**

Por ejemplo:

def68337867cb4f1f95b6b811fedbfccdd7780d10a95cc072077088ea + 746F6B656E

## Minting Policy : partes de un token

---

Un token está compuesto por dos partes: **Policy Id + Token Name**

Por ejemplo:

def68337867cb4f1f95b6b811fedbfccdd7780d10a95cc072077088ea + 746F6B656E



La **Policy Id** es el Hash  
del código en Plutus  
utilizado para crear  
el token

## Minting Policy : partes de un token

---

Un token está compuesto por dos partes: **Policy Id + Token Name**

Por ejemplo:

def68337867cb4f1f95b6b811fedbfccdd7780d10a95cc072077088ea + 746F6B656E



El **Token Name** es el nombre del token representado en formato hexadecimal

## **Minting Policy : Utilidades con MeshJS**

---

Funciones útiles para trabajar con Minting policies:

## Minting Policy : Utilidades con MeshJS

---

Funciones útiles para trabajar con Minting policies:

`resolveScriptHash(`Script, Version`)` : Función para obtener Policy Id dado un Script de plutos y la versión del plutos script.

## Minting Policy : Utilidades con MeshJS

---

Funciones útiles para trabajar con Minting policies:

`resolveScriptHash(`Script, Version`)` : Función para obtener Policy Id dado un Script de plutos y la versión del plutos script.

`mintAsset(`Script, Mint, Redeemer`)` : Método de la clase Transaction para minar tokens.

## Minting Policy : Utilidades con MeshJS

---

Funciones útiles para trabajar con Minting policies:

`resolveScriptHash(`Script, Version`)` : Función para obtener Policy Id dado un Script de plutos y la versión del plutos script.

`mintAsset(`Script, Mint, Redeemer`)` : Método de la clase Transaction para minar tokens.

`mint(`quantity, policy, name`)` : Método de la clase MeshTxBuilder para minar tokens.

## Minting Policy : Utilidades con MeshJS

---

Funciones útiles para trabajar con Minting policies:

`resolveScriptHash(`Script, Version`)` : Función para obtener Policy Id dado un Script de plutos y la versión del plutos script.

`mintAsset(`Script, Mint, Redeemer`)` : Método de la clase Transaction para minar tokens.

`mint(`quantity, policy, name`)` : Método de la clase MeshTxBuilder para minar tokens.

`mintPlutusScriptV3()` : Especifica que el Script de Plutus corresponde a la versión 3.

## Minting Policy : Utilidades con MeshJS

---

Funciones útiles para trabajar con Minting policies:

`resolveScriptHash( Script, Version )` : Función para obtener Policy Id dado un Script de plutos y la versión del plutos script.

`mintAsset( Script, Mint, Redeemer )` : Método de la clase Transaction para minar tokens.

`mint( quantity, policy, name )` : Método de la clase MeshTxBuilder para minar tokens.

`mintPlutusScriptV3()` : Especifica que el Script de Plutus corresponde a la versión 3.

`mintRedeemerValue( redeemer )` : Especifica el valor de redeemer para ejecutar el código de Plutus.

## Minting Policy : Utilidades con MeshJS

---

Funciones útiles para trabajar con Minting policies:

`resolveScriptHash(`Script, Version`)` : Función para obtener Policy Id dado un Script de plutos y la versión del plutos script.

`mintAsset(`Script, Mint, Redeemer`)` : Método de la clase Transaction para minar tokens.

`mint(`quantity, policy, name`)` : Método de la clase MeshTxBuilder para minar tokens.

`mintPlutusScriptV3()` : Especifica que el Script de Plutus corresponde a la versión 3.

`mintRedeemerValue(`redeemer`)` : Especifica el valor de redeemer para ejecutar el código de Plutus.

`mintingScript(`Script`)` : Especifica el código de Plutus que se va a ejecutar.

## Minting Policy : Tipos de reglas

---

Se pueden programar diferentes reglas para minar tokens. Algunos ejemplos comunes:

## Minting Policy : Tipos de reglas

---

Se pueden programar diferentes reglas para minar tokens. Algunos ejemplos comunes:

- ¿Cuántos tokens se pueden crear?

## Minting Policy : Tipos de reglas

---

Se pueden programar diferentes reglas para minar tokens. Algunos ejemplos comunes:

- ¿Cuántos tokens se pueden crear?
- ¿Cuál nombre deben de tener los tokens?

## Minting Policy : Tipos de reglas

---

Se pueden programar diferentes reglas para minar tokens. Algunos ejemplos comunes:

- ¿Cuántos tokens se pueden crear?
- ¿Cuál nombre deben de tener los tokens?
- ¿Cuándo se pueden crear los tokens?

## Minting Policy : Tipos de reglas

---

Se pueden programar diferentes reglas para minar tokens. Algunos ejemplos comunes:

- ¿Cuántos tokens se pueden crear?
- ¿Cuál nombre deben de tener los tokens?
- ¿Cuándo se pueden crear los tokens?
- ¿Quién puede crear los tokens?

## Minting Policy : Tipos de reglas

---

Se pueden programar diferentes reglas para minar tokens. Algunos ejemplos comunes:

- ¿Cuántos tokens se pueden crear?
- ¿Cuál nombre deben de tener los tokens?
- ¿Cuándo se pueden crear los tokens?
- ¿Quién puede crear los tokens?
- ¿Se debe correr un Spend Validator para crear un token?

## Minting Policy : Ejercicios

---

- **Token firmado:** Crear un script que permita crear un token sólamente cuando una persona específica firme la transacción.

## Minting Policy : Ejercicios

---

- **Token firmado:** Crear un script que permita crear un token sólamente cuando una persona específica firme la transacción.
- **Token firmado y selección de token name:** Crear un script que permita crear un token sólamente cuando una persona específica firme la transacción y además el nft tiene un nombre específico.

## Minting Policy : Ejercicios

---

- **Token firmado:** Crear un script que permita crear un token sólamente cuando una persona específica firme la transacción.
- **Token firmado y selección de token name:** Crear un script que permita crear un token sólamente cuando una persona específica firme la transacción y además el nft tiene un nombre específico.

### *i* Info

- Los Scripts deben ser parametrizados
- Escribir código on-chain y off-chain

## NFT : ¿Qué son?

---

- Un **NFT (Non-Fungible Token)** es un token único e irrepetible.



## NFT : ¿Qué son?

---



- Un **NFT (Non-Fungible Token)** es un token único e irrepetible.
- Solamente se puede crear uno de cada NFT en **toda la historia de la blockchain**.

# NFT : ¿Qué son?

---



- Un **NFT (Non-Fungible Token)** es un token único e irrepetible.
- Solamente se puede crear uno de cada NFT en **toda la historia de la blockchain**.
- Esto se puede lograr por medio de diferentes técnicas utilizando código Plutus (Aiken).

## NFT : ¿Para qué sirven?

---

Los NFTs pueden tener varios usos por su naturaleza irrepetible:



## NFT : ¿Para qué sirven?

---

Los NFTs pueden tener varios usos por su naturaleza irrepetible:

- **Representar** la propiedad de algún objeto físico.



## NFT : ¿Para qué sirven?

---

Los NFTs pueden tener varios usos por su naturaleza irrepetible:

- **Representar** la propiedad de algún objeto físico.
- **Representar** los derechos de autor de alguna obra.



## NFT : ¿Para qué sirven?

---



Los NFTs pueden tener varios usos por su naturaleza irrepetible:

- **Representar** la propiedad de algún objeto físico.
- **Representar** los derechos de autor de alguna obra.
- **Representar** la autoridad para poder realizar una acción dentro de la blockchain

# NFT : ¿Para qué sirven?

---



Los NFTs pueden tener varios usos por su naturaleza irrepetible:

- **Representar** la propiedad de algún objeto físico.
- **Representar** los derechos de autor de alguna obra.
- **Representar** la autoridad para poder realizar una acción dentro de la blockchain
- ...

## **NFT : Qué tiene que hacer el contrato?**

---

Una de las técnicas más utilizadas en Cardano para la creación de NFTs es el uso de Validadores parametrizados:

## NFT : Qué tiene que hacer el contrato?

---

Una de las técnicas más utilizadas en Cardano para la creación de NFTs es el uso de Validadores parametrizados:

- **UTxO como parámetro:** Se parametriza el validador con un UTxO existente y se verifica que este se consuma en la transacción en la que se crea el NFT.

## NFT : Qué tiene que hacer el contrato?

---

Una de las técnicas más utilizadas en Cardano para la creación de NFTs es el uso de Validadores parametrizados:

- **UTxO como parámetro:** Se parametriza el validador con un UTxO existente y se verifica que este se consuma en la transacción en la que se crea el NFT.
- **Solo crear un token:** En la lógica del script se estipula que se quiere minar sólamente un token. Asegurándonos de que este no se repita.

## NFT : Qué tiene que hacer el contrato?

---

Una de las técnicas más utilizadas en Cardano para la creación de NFTs es el uso de Validadores parametrizados:

- **UTxO como parámetro:** Se parametriza el validador con un UTxO existente y se verifica que este se consuma en la transacción en la que se crea el NFT.
- **Solo crear un token:** En la lógica del script se estipula que se quiere minar sólamente un token. Asegurándonos de que este no se repita.
- **Token name:** También se puede decidir por el nombre del token. (Opcional).

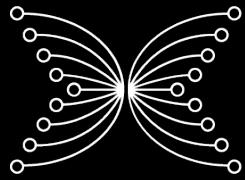
## NFT : Qué tiene que hacer el contrato?

---

Una de las técnicas más utilizadas en Cardano para la creación de NFTs es el uso de Validadores parametrizados:

- **UTxO como parámetro:** Se parametriza el validador con un UTxO existente y se verifica que este se consuma en la transacción en la que se crea el NFT.
- **Solo crear un token:** En la lógica del script se estipula que se quiere minar sólamente un token. Asegurándonos de que este no se repita.
- **Token name:** También se puede decidir por el nombre del token. (Opcional).

Si se trata de crear un nuevo NFT con el mismo Script, se debe utilizar un nuevo UTxO. Por lo tanto, el Policy ID cambiará.



INPUT | OUTPUT

# Preguntas?

