Approbation Token Gating dApp T.P  
Julián Vázquez SIB-2025s1 UNQ

# 1. Objetivo General

Desarrollar una aplicación descentralizada (dApp) que implemente un sistema de token gating educativo basado en el estándar ERC-1155, permitiendo dos flujos diferenciados: uno para estudiantes y otro para docentes. Se validan NFTs previos para permitir la emisión y recepción de un segundo NFT que acredita la aprobación de la cursada.

# 2. Actores del Sistema

|  |  |
| --- | --- |
| Alumno | Se conecta con su wallet, presenta NFTs previos y emite un NFT de TP. Recibe, si se dan las condiciones, un NFT de acreditación de aprobación de cursada. |
| Docente | Recibe NFTs de los alumnos. Puede emitir un NFT de promoción. |

# 3. Flujos Funcionales

## Flujo 1 – Alumno

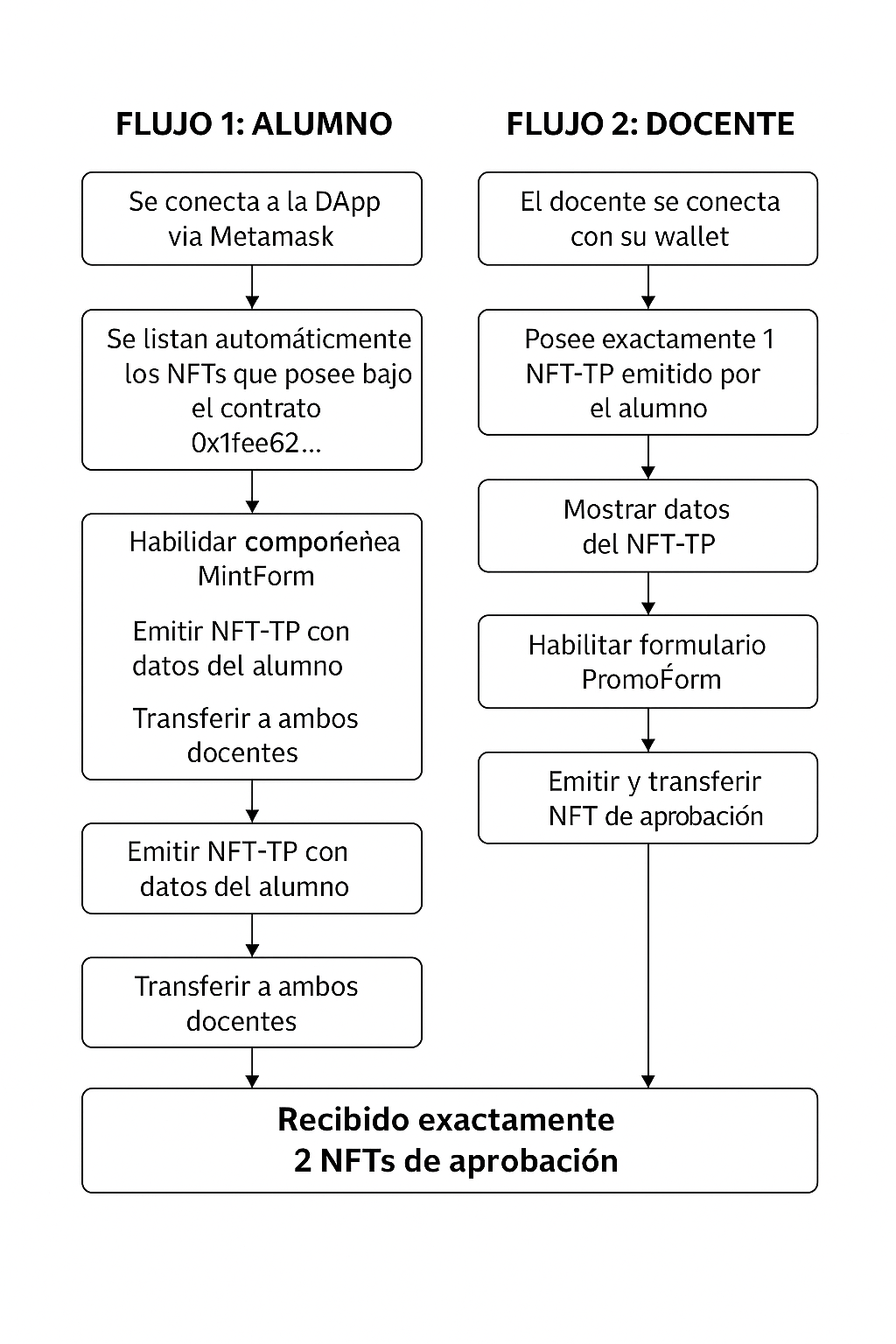
1. Se conecta a la dApp vía Metamask.  
2. Se listan sus NFTs bajo contrato 0x1fee62...f91d.  
3. Se validan 3 condiciones:  
 - Posee al menos 10 NFTs del contrato.  
 - Fueron recibidos antes del 28/04/2025.  
 - No fueron transferidos:  
 \* Fue recibido desde address(0), lo que indica que fue minteado directamente. \* No debe existir ningún evento posterior donde ese NFT haya salido de la wallet.  
4. Si se cumplen las 3 condiciones, se habilita un botón para emitir y transferir un **NFT-TP** automáticamente a ambos docentes en una sola transacción (mint + transfer).

## Flujo 2 – Docente

1. Se conecta con wallet validada que pertenece a los docentes, caso contrario no se accede a esta parte.  
2. Se verifica que el docente posea al menos un NFT-TP que haya sido emitido por el alumno cuya wallet haya sido ingresada como input.  
3. Si la validación es correcta:  
 - Visualiza el NFT-TP recibido y sus datos on-chain.  
4. Se habilita el formulario para emitir un NFT de promoción personalizado.  
 - El NFT incluye campos on-chain: alumno, nota, comentario, y se transfiere a la wallet del alumno mediante un botón que hace el mint+transfer del mismo.

## Flujo 3 – Alumno Aprobado

El alumno podrá visualizar su NFT de aprobación solo si recibió exactamente 2 NFTs de promoción (uno de cada profesor). Caso contrario, se muestra la pantalla correspondiente al flujo 1 en estado que se encuentre el alumno.



# 4. Infraestructura Técnica

#### FRONTEND

* React + Vite + Tailwind.
* Ethers.js para conexión Web3.
* ABI del contrato ERC-1155.
* Componente <NFTCard> reutilizable para visualización de ambos tipos de NFT

#### BACKEND

# No se requiere backend persistente. Toda la lógica de emisión, validación y consulta de datos está contenida on-chain en los contratos inteligentes.

# 5. Smart Contracts

Dos contratos ERC-1155

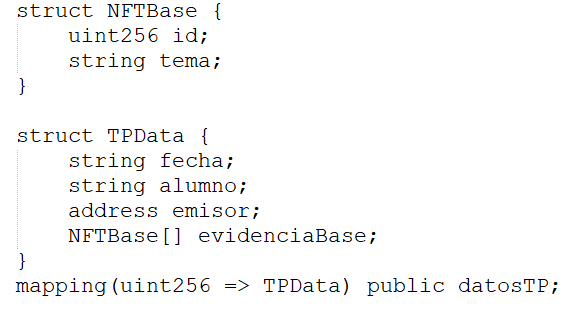
|  |  |
| --- | --- |
| TPNFT | NFT emitido por alumno a profesores |
| PROMONFT | NFT emitido por profesor al alumno |

#### Estructuras y funciones requeridas (TPNFT): Metadata fija

* name: Título genérico del NFT.
* description: Breve descripción del uso del NFT como evidencia de entrega de trabajo.
* image: Imagen institucional o representativa del TP.

**Datos variables on-chain:**

* fecha: Fecha de emisión del NFT.
* alumno: Nombre del alumno.
* emisor: Wallet que emitió el NFT.
* PoF: La lista de los id y tema de los 10 nft que me acreditaron a emitirlo.

****

##### **Funciones del estándar ERC-1155:**

* balanceOf(address, uint256)
  + Devuelve cuántos tokens del tipo id tiene la cuenta account.
* uri(uint256 tokenId)
  + Devuelve la URI del archivo JSON que contiene la metadata del token

##### **Funciones propias:**

function mintAndTransfer(address profe1, address profe2, TPData calldata datos) external onlyOwner  
Emite (mint) un nuevo NFT-TP asociado a un tokenId único.  
Guarda en el contrato los datos variables on-chain correspondientes a ese tokenId.  
Transfiere (transfer) automáticamente ese NFT a dos wallets docentes.  
Impide que un alumno emita más de un NFT-TP si ya lo hizo antes.

function datosDeAsist(uint256 tokenId) public view returns (string memory fecha, string memory alumno, address emisor, …)  
Devuelve los datos variables on-chain del token especificado.  
  
**Restricciones logicas del contrato:**

Cada alumno puede emitir un único NFT-TP por profesor. Esto significa que:

* Un alumno puede emitir un NFT a Dani una sola vez.
* Un alumno puede emitir un NFT a Pablo una sola vez.
* Cada emisión resulta en un tokenId distinto, pero controlado y validado por contrato.
* Solo el alumno puede emitir su propio NFT

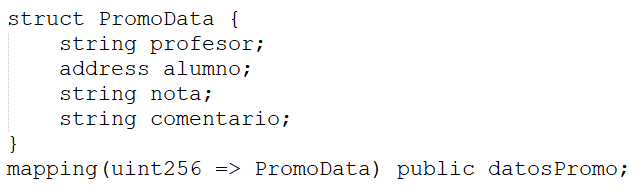
**Estructuras y funciones requeridas (PROMONFT):**

#### Metadata fija:

* name: "NFT-Promoción"
* description: Acreditación oficial de aprobación.
* image: Imagen fija de certificación.

#### Datos variables on-chain:

* profesor: Nombre del profesor que emitió el NFT.
* alumno: Wallet del alumno que lo recibe.
* nota: Nota numérica o textual otorgada.
* comentario: Comentario evaluativo adicional.

**

##### **Funciones del estándar ERC-1155:**

* balanceOf(address, uint256)
  + Se usa desde el frontend para validar posesión de NFTs (profesores y alumnos).
* uri(uint256 tokenId)
  + Devuelve la URI del archivo JSON que contiene la metadata del token

##### **Funciones propias:**

function mintPromo(address to, PromoData calldata datos) external onlyOwnerEmite el NFT de promoción con los datos variables, asociándolo al alumno.

function datosDePromo(uint256 tokenId) public view returns (address profesor, address alumno, string memory nota, string memory comentario)Devuelve los datos evaluativos asociados al NFT.

**Restricciones lógicas del contrato:**

* Valida que el emisor sea un profesor autorizado.
* Verifica que ese profesor aún no emitió un NFT de promoción para ese alumno.

**6. Componentes**

|  |  |
| --- | --- |
| Header | Encabezado fijo con título y conexión Metamask. |
| NFTList | Cards que muestran NFTs poseídos por wallet conectada. |
| CheckPanel | Valida condiciones A, B, C para permitir emisión. |
| MintForm | Formulario de emisión de NFT-TP por parte del alumno. |
| ProfessorPanel | Sección exclusiva para Dani y Pablo. |
| PromoForm | Formulario de emisión del NFT de promoción. |