**Arquitectura sugerida (MVP prolijo)**

**1) InstitutionRegistry (control de emisores)**

**Responsabilidad:** llevar quién puede emitir.

* grantIssuer(address account) — solo Admin.
* revokeIssuer(address account) — solo Admin.
* isIssuer(address account) → bool — vista pública.
* getAllIssuers() → address[] (opcional; mejor eventos + indexador).  
  **Eventos:**
* IssuerGranted(address issuer)
* IssuerRevoked(address issuer)

Podés implementar Admin con AccessControl (DEFAULT\_ADMIN\_ROLE) acá. El SoulboundCertificate consulta a este contrato para validar emisores.

**2) SoulboundCertificate (el SBT en sí)**

**Responsabilidad:** emitir, guardar y revocar certificados.

* issueCertificate(address holder, …meta…) → uint256 tokenId — solo emisor autorizado (consulta a InstitutionRegistry).
* revoke(uint256 tokenId) — solo la **misma institución** que lo emitió.
* getCertificate(uint256 tokenId) → Certificate — vista.
* locked(uint256 tokenId) → bool — siempre true (EIP-5192 minimal, opcional).
* **Bloquear transferencias y approvals** (override de transferFrom/safeTransferFrom/approve/setApprovalForAll → revert).  
  **Eventos:**
* Issued(uint256 tokenId, address issuer, address holder)
* Revoked(uint256 tokenId, address issuer)

Este contrato puede heredar de ERC721 (OpenZeppelin). El tokenId lo autoincrementás.

**3) VerifierPortal (opcional, solo lectura)**

**Responsabilidad:** facilitar consultas para terceros (sin permisos).

* verify(uint256 tokenId) → (issuer, holder, revoked, cid, contentHash, issueDate)
* holderCertificates(address holder, uint256 offset, uint256 limit) → uint256[] (si mantenés índice; ver nota abajo).
* issuerCertificates(address issuer, uint256 offset, uint256 limit) → uint256[] (idem).

Este contrato no es imprescindible. Es útil si querés **APIs de lectura cómodas** sin exponer toda la estructura interna.

**4) (Opcional) InstitutionMetadata**

**Responsabilidad:** metadatos de instituciones (nombre, metadataCID, estado).

* Útil si querés algo más que un booleano de “es emisor”.
* Si no, mantenelo fuera del chain o adentro de InstitutionRegistry.

**Índices y listados (importante)**

* **Evitar ERC721Enumerable** en mainnets por costo de gas.
* Alternativas:
  + **The Graph** / indexador off-chain (lo más profesional).
  + Mantener **mappings ligeros** y paginables:
    - mapping(address => uint256[]) byHolder;
    - mapping(address => uint256[]) byIssuer;
  + Si hacés esto, exponé funciones **paginadas** (offset/limit).
  + No borres IDs al revocar; **marcá estado** y filtrá en la lectura.

Guardar una lista certificates dentro de un Student estructuralmente es lo mismo que byHolder[wallet]. Preferí el mapping directo; menos acoplamiento y más simple.