Integration of web3 API with study control system

Web3 API integration with the study control system refers to the process of connecting blockchain technology with the study management system to record and verify applicant information securely and transparently.

Advantages
1. Transparency: The blockchain provides an immutable and transparent blockchain, which ensures that applicant information is accurate and cannot be altered.
2 Security: Blockchain technology uses advanced cryptography to protect applicant information, reducing the risk of unauthorized access or data loss.
3. Immutability: The blockchain ensures that applicant information is immutable, meaning that once registered, it cannot be modified or deleted.
4. Automation: The integration of the web3 API with the study control system allows automating processes, such as the verification of applicant information, which reduces time and manual effort.

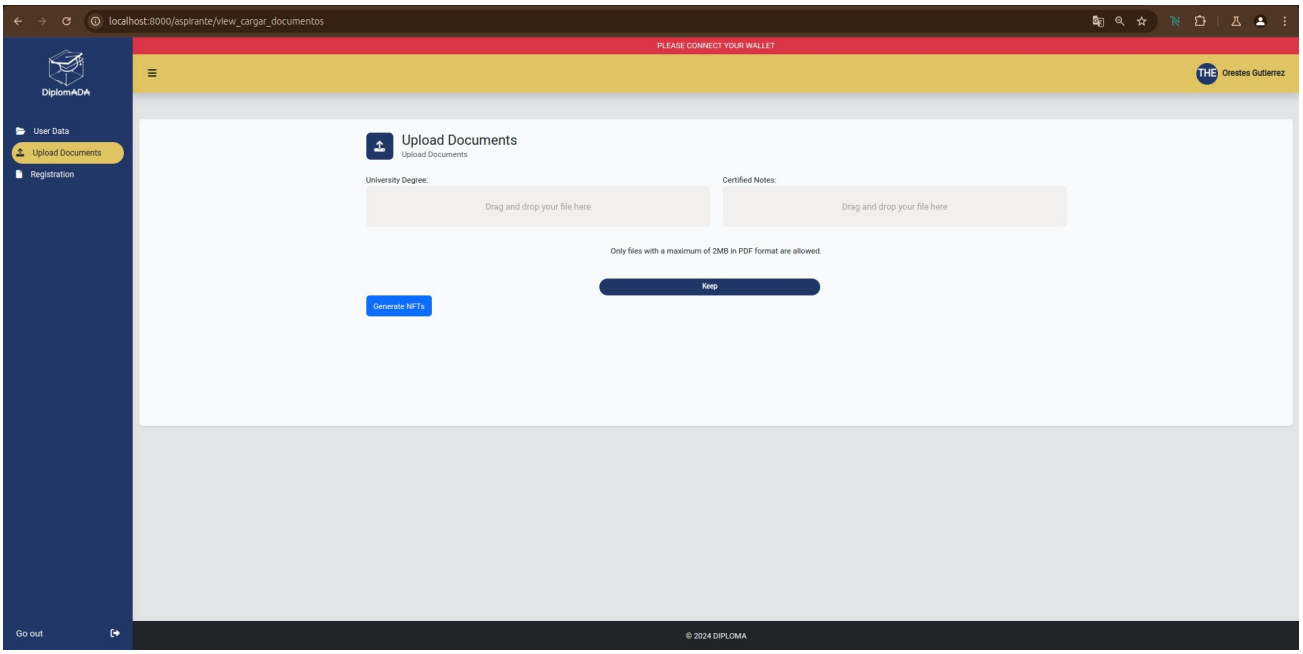
Architecture
The architecture of the web3 API integration with the study control system consists of the following components:
1. web3 API: The web3 API is responsible for interacting with the blockchain and recording applicant information in the smart contract.
2. Study control system: The study control system is responsible for managing the applicants' information and verifying their authenticity.
3. Integration service: The integration service is responsible for connecting the web3 API with the study control system and automating the registration and verification processes.

Workflow
The workflow of integrating the web3 API with the study control system is described below:
1. The applicant provides his/her personal information and required documents.
2. The applicant's information is sent to the study control system for verification.
3. The study control system verifies the applicant's information and sends it to the integration service.
4. The integration service uses the web API3 to register the applicant's information in the smart contract.
5. The web3 API returns the transaction hash to the integration service.
6. The integration service returns the result of the transaction to the study control system.
7. The study control system updates the applicant information with the result of the transaction.

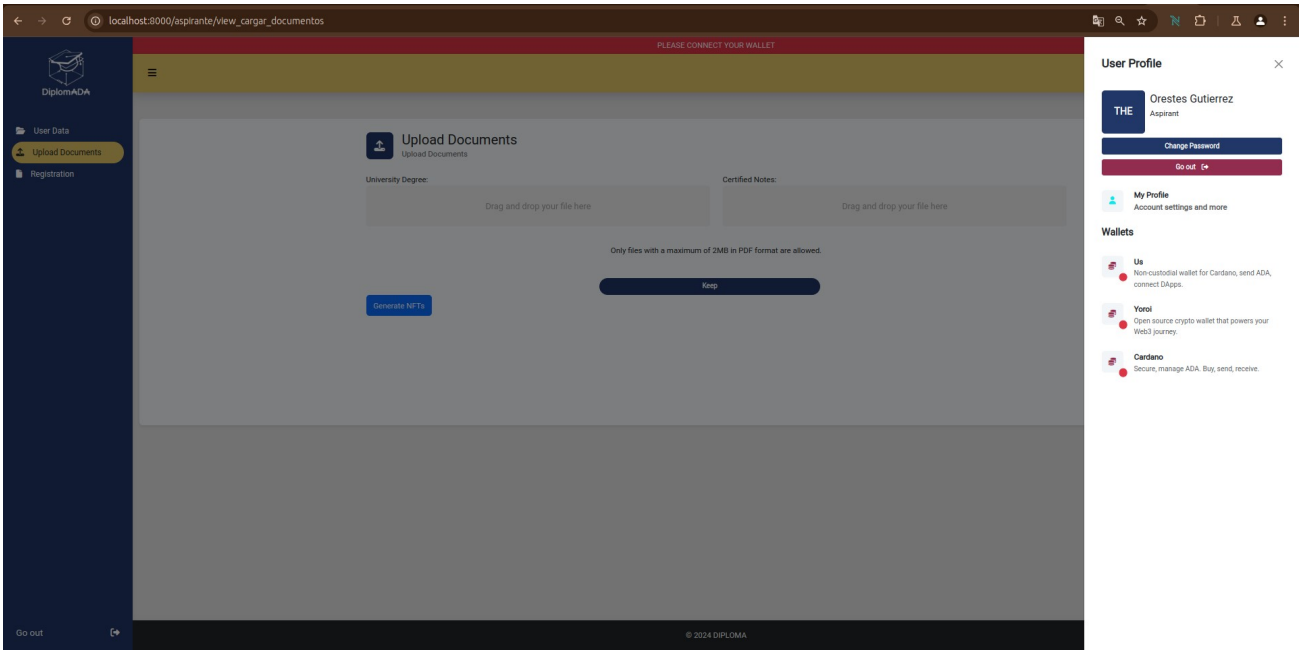
Benefits
The integration of the web3 API with the study control system offers several benefits, such as:
1. improves transparency and security: blockchain provides an immutable and transparent blockchain, which ensures the accuracy and security of applicant information.
2. Reduces time and manual effort: Process automation reduces the time and manual effort required to verify applicant information.
3 Improved user experience: The integration of the web API3 with the study control system provides a faster and more secure user experience.
In summary, the integration of the web3 API with the study control system is an innovative solution that combines blockchain technology with the study management system to record and verify applicant information in a secure and transparent way.

For this, we will show the process of connecting and interacting with the NAMI wallet that will allow us to perform transactions within the platform on blockchain:

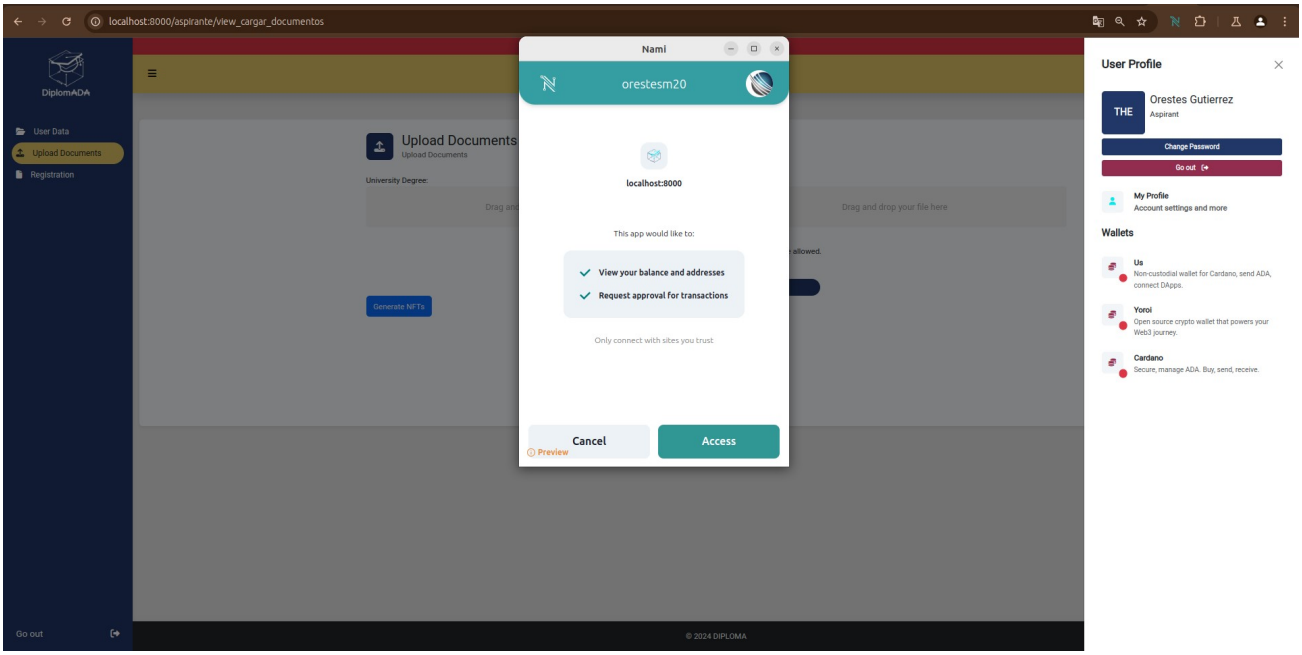
NAMI wallet disconnected application.



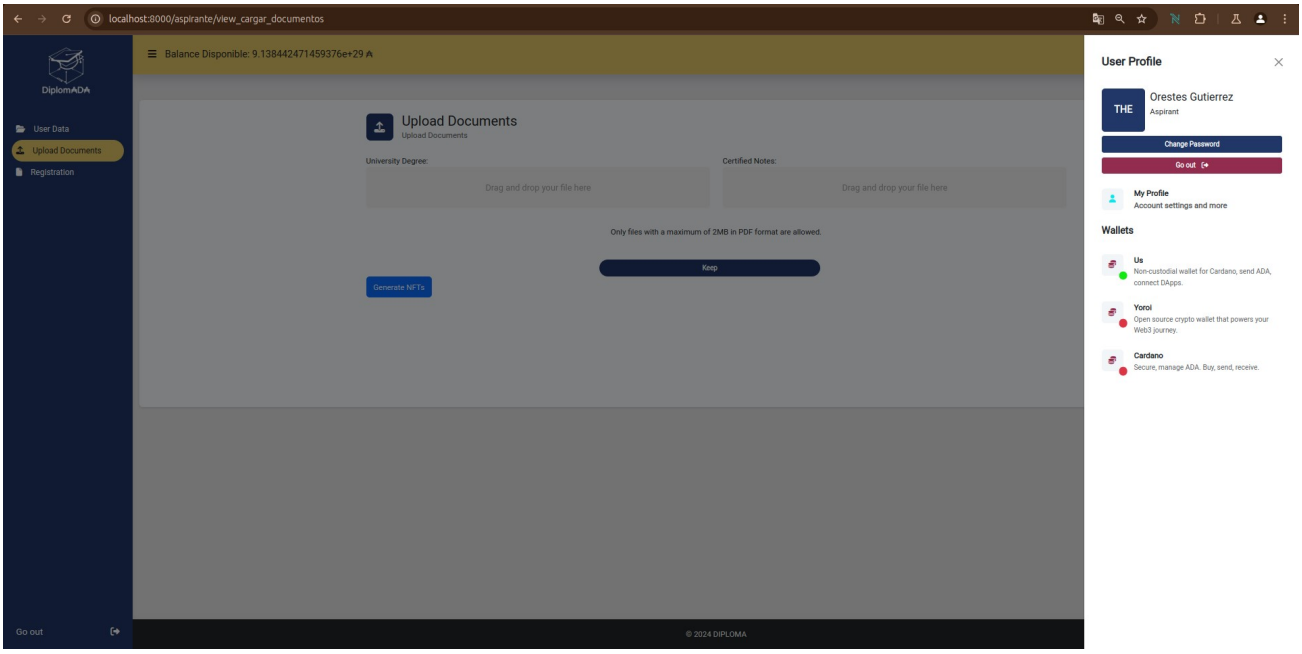
Request for connection to NAMI wallet



Signature of the wallet connection request



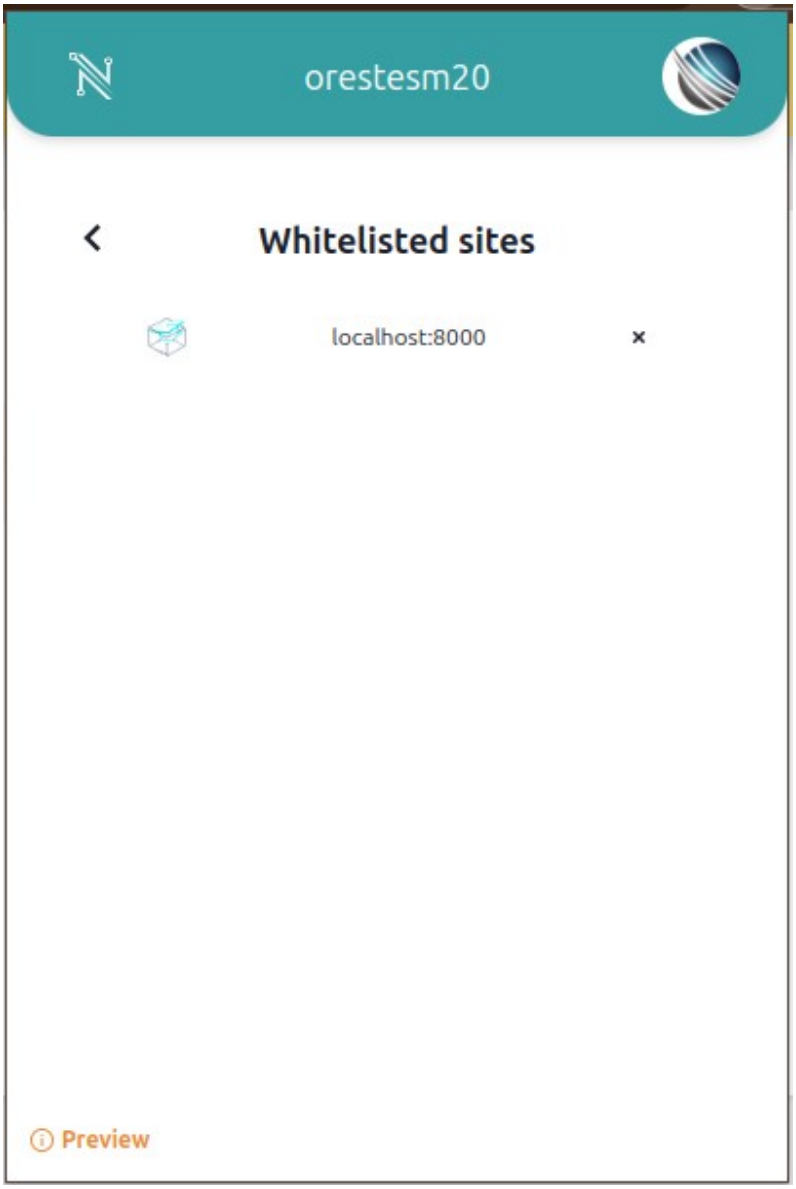
Application connected to NAMI Wallet



NAMI wallet connection function of wallet.ts file

```
import { cardanoIsEnabled, conectar, getBalance } from'./demostra';
constwallet=document.getElementById('wallet-1');

wallet?.addEventListener('click', async () => {
awaitconectar();
awaitgetBalance();
//console.log('llegue a conectar',await conectar());
});
```

NAMI Wallet Connection Verification



Code snippet of Diplomada web application integration with API for blockchain registration.

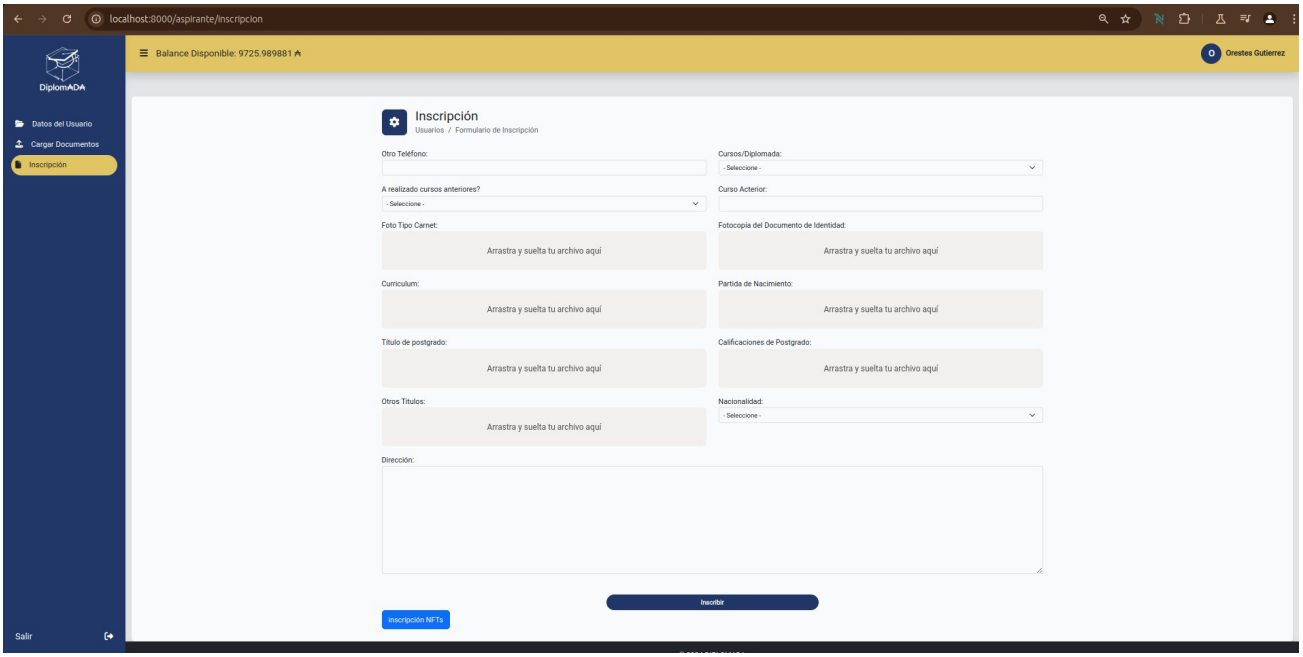
```
exportconstcrearTitulos=async ():Promise<Resultado<any>>=> {
//let resultado: Resultado<any> = { type: "error", error: new Error("No se pudo crear el titulo") };
try {cardanoIsEnabled
if (lucid) {
console.log("Creando NFT de Titulos y Notas ...");
constdadaPM_TitulosyNotas:MintingPolicy= {
type:"PlutusV3",
script:contratos.scripts.pm_titulosNotas
}

constdadaVal_Validacion:SpendingValidator= {
type:"PlutusV3",
script:contratos.scripts.val_verificacion
}

//console.log("cree las variables de los contratos V3.")
constdadaValDireccion:string=validatorToAddress("Preview",dadaVal_Validacion);
//console.log("pase validtortoAddress");
constdireccionEstudiante:Address=awaitlucid.wallet().address();
consttokenName_Titulos="dtitulo0917";
consttokenName_Notas="dADANotas0917";
consttitulos_tokenName=fromText(tokenName_Titulos);
constnotas_tokenName=fromText(tokenName_Notas);
consttitulosynotas_pid:PolicyId=mintingPolicyToId(dadaPM_TitulosyNotas);
consttitulos_dada:Unit=toUnit(titulosynotas_pid, titulos_tokenName);
constnotas_dada:Unit=toUnit(titulosynotas_pid, notas_tokenName);
consttitulosyNotasRedeemer=BigInt(1);
constmintRedeemer=Data.to(titulosyNotasRedeemer);

// console.log(dadaPM_TitulosyNotas);
// console.log("validator address ->" + dadaValDireccion);
// console.log("address ->" + direccionEstudiante);
// console.log("Nombre Token Titulos: " + tokenName_Titulos + " ->" + titulos_tokenName);
// console.log("Nombre Token Notas: " + tokenName_Notas + " ->" + notas_tokenName);
// console.log("pid ->" + titulosynotas_pid)
// console.log("NFT Titulos ->" + titulos_dada);
// console.log("NFT Notas ->" + notas_dada);
// console.log(fromUnit(titulos_dada));
// console.log(fromUnit(notas_dada));
// console.log("redeeemer ->" + titulosyNotasRedeemer);
// console.log(mintRedeemer);


constjsonData:MD_Titulos= {
[titulosynotas_pid]: {
```

```
[tokenName_Titulos]: {
id:1,
name:"Diplomada Titulo Certificado 22",
image:"https://shorturl.at/NRvjj",
description:"Diplomada Titulo Test"
},
[tokenName_Notas]: {
id:1,
name:"Diplomada Notas Certificadas 22",
image:"https://shorturl.at/NRvjj",
description:"Diplomada Notas Test"
},
"datos_estudiante": {
hash:"fac7b8513f4b985174e88a02ee8165fc",
nombres:"Roberto Jose",
apellidos:"Cerrud Campos"
},
},
};
constdatum_crudo=Data.to(BigInt(1));
console.log("Datum crudo: ->"+datum_crudo);
constdatum:OutputDatum= {kind:"inline", value:datum_crudo};

consttx=awaitlucid
.newTx()
.mintAssets({
[titulos_dada]:BigInt(2),
[notas_dada]:BigInt(2),
}, mintRedeemer)
.pay.ToAddress(direccionEstudiante, { [titulos_dada]:BigInt(1), [notas_dada]:BigInt(1), lovelace:BigInt(2000000)})
.pay.ToContract(dadaValDireccion, datum , { [titulos_dada]:BigInt(1), [notas_dada]:BigInt(1),
lovelace:BigInt(2000000)})
.attach.MintingPolicy(dadaPM_TitulosyNotas)
.attachMetadata("721", jsonData)
.complete();


constsignedTx=awaittx.sign.withWallet().complete();
console.log('signedTX '+signedTx);
consttxHash=awaitsignedTx.submit();
console.log("txHash ->"+txHash);
constsuccess=awaitlucid!.awaitTx(txHash)
console.log("Funciono ->"+success);
constmensaje="por aqui si es";
console.log(mensaje);

return { type:"ok", data:txHash };
```

```
}
} catch (error) {
console.log("error ->"+error);
if (errorinstanceofError) return { type:"error", error:error };
return { type:"error", error:newError(errorasstring) };
}
}
```

In this part of the code the mining request for the creation of the NFTs of the Degree and Certified grades is performed, additionally the registration in the blockchain is performed, this interaction generates a transaction hash that can be verified in CARDANO's block explorer.

Like the degree and grades, the enrollment process also performs the transaction where all the information of the students is recorded at the time of making their enrollment process, this transaction results in the creation of a NFTs and the registration in the blockchain.



This is the fraction of code where the process of interaction with the smart contract, the creation of the NFT and the registration of the blockchain corresponding to the enrollment process is shown.

```
consthastala=document.getElementById("hastala");
if (hastala) {
hastala.addEventListener("click", asyncfunction () {
console.log(datos);
vardemo=crearInscripcion(datos);

console.log('este',(awaitdemo).type);
//console.log('este');
if ((awaitdemo).type=='error') {
alert('debe Reconectar la billetera');
```

```
}
if ((awaitdemo).type=='ok') {
alert('Proceso de Inscripcion completado');
}
//colocar las notificaciones verificando demo en el campo type
//await crearTitulos();
});
}


exportconstcrearInscripcion=async (params:ParamsInscripcion):Promise<Resultado<any>>=> {
try {cardanoIsEnabled
if (lucid) {
//console.log("Creando NFT de Inscripcion ...");
constdadaPM_Inscripcion:MintingPolicy= {
type:"PlutusV3",
script:contratos.scripts.pm_incripcion
}
constdadaVal_Validacion:SpendingValidator= {
type:"PlutusV3",
script:contratos.scripts.val_verificacion
}
//console.log("cree las variables de los contratos V3.")
constdadaValDireccion:string=validatorToAddress("Preview",dadaVal_Validacion);
//console.log("pase validtortoAddress");
constdireccionEstudiante:Address=awaitlucid.wallet().address();
consttokenName_Inscripcion=dumyParams.cedula;
constnftInscripcion_tokenName=fromText(tokenName_Inscripcion);
constnftInscripcion_pid:PolicyId=mintingPolicyToId(dadaPM_Inscripcion);
constinscripcion_dada:Unit=toUnit(nftInscripcion_pid, nftInscripcion_tokenName);
constinscripcionRedeemer=BigInt(1);
constmintRedeemer=Data.to(inscripcionRedeemer);
// console.log(dadaPM_TitulosyNotas);
// console.log("validator address ->" + dadaValDireccion);
// console.log("address ->" + direccionEstudiante);
// console.log("Nombre Token Titulos: " + tokenName_Titulos + " ->" + titulos_tokenName);
// console.log("Nombre Token Notas: " + tokenName_Notas + " ->" + notas_tokenName);
// console.log("pid ->" + titulosynotas_pid)
// console.log("NFT Titulos ->" + titulos_dada);
// console.log("NFT Notas ->" + notas_dada);
// console.log(fromUnit(titulos_dada));
// console.log(fromUnit(notas_dada));
// console.log("redeeemer ->" + titulosyNotasRedeemer);
// console.log(mintRedeemer);
constjsonData:MD_Titulos= {
[nftInscripcion_pid]: {
[tokenName_Inscripcion]: {
```

```
id:1,
name:"Diplomada Inscripcion 1",
image: [params.url],
description:"Diplomada Titulo Test"
},
"datos_estudiante": {
hash:"fac7b8513f4b985174e88a02ee8165fc",
nombres: [params.nombres],
apellidos: [params.apellidos],
cedula: [params.cedula],
sexo: [params.sexo],
fecha_nac: [params.fecha_nac],
direccion: [params.direccion],
telefono_habitacion: [params.telefono_habitacion],
telefono_otros: [params.telefono_otros],
celular: [params.celular],
correo: [params.correo],
curso:"Diplomada en Desarrollo de Aplicacopmes en Cardano"
},
},
};
constdatum_crudo=Data.to(BigInt(1));
//console.log("Datum crudo: ->" + datum_crudo);
constdatum:OutputDatum= {kind:"inline", value:datum_crudo};
consttx=awaitlucid
.newTx()
.mintAssets({[inscripcion_dada]:BigInt(2)}, mintRedeemer)
.pay.ToAddress(direccionEstudiante, { [inscripcion_dada]:BigInt(1), lovelace:BigInt(2000000)})
.pay.ToContract(dadaValDireccion, datum , { [inscripcion_dada]:BigInt(1), lovelace:BigInt(2000000)})
.attach.MintingPolicy(dadaPM_Inscripcion)
.attachMetadata("721", jsonData)
.complete();
constsignedTx=awaittx.sign.withWallet().complete();
consttxHash=awaitsignedTx.submit();
console.log("txHash ->"+txHash);
constsuccess=awaitlucid!.awaitTx(txHash)
console.log("Funciono ->"+success);
constmensaje="por aqui si es";
console.log(mensaje);
return { type:"ok", data:txHash };
}
} catch (error) {
console.log("error ->"+error);
if (errorinstanceofError) return { type:"error", error:error };
return { type:"error", error:newError(errorasstring) };
}
}
```

Conclusion

We can conclude that this proves that the process of interaction with the service is fully functional and also allows us to perform the registration effectively on the blockchain, using the lucid platform and the standard CIP30 protocol as a means of interaction for the creation of the NFTs requested by the application.