

| Asignatura | Datos del alumno | Fecha |
|----------------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Trabajo Final de Experto



Martí Prat Mayans

Desarrollo de Aplicaciones Blockchain

2022

| Asignatura | Datos del alumno | Fecha |
|----------------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Índice

| | |
|--|-----------|
| Sistema de Trazabilidad de Procesos Productivos | 3 |
| Roles | 3 |
| Justificación del uso de la Tecnología Blockchain | 4 |
| Diagrama de Despliegue | 5 |
| Instrucciones de Despliegue | 7 |
| Manual de Usuario | 11 |
| Pruebas – Main | 13 |
| Conclusiones | 19 |

| Asignatura | Datos del alumno | Fecha |
|---------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Sistema de Trazabilidad de Procesos Productivos

El sistema que he diseñado se basa en un problema que tienen las empresas actualmente a la hora de hacer toda la trazabilidad de los procesos internos de producción que tienen.

Actualmente, con toda la situación macroeconómica actual a cualquier fabricante se le hace muy difícil controlar el tiempo de producción de cualquier proceso, con lo que su cliente final, muchas veces no sabe en qué estado está su pedido, con lo que tenemos un problema de comunicación y malentendidos entre el fabricante y el cliente.

Para solucionar esto, he implementado una solución basada en Blockchain que permite saber el estado de cada pedido del cliente y, así es más fácil tanto para el cliente saber el estado y el tiempo aproximado que tardará en tener el pedido listo para entregar-se.



Roles

1. **Rol Empresa:** la empresa es la que se encarga de añadir los procesos de producción en la web y, gestionar las diferentes etapas del proceso para que el cliente este actualizado en cada momento.
2. **Rol Cliente:** el cliente lo único que puede hacer es consultar el estado de su pedido y saber la etapa en la que se encuentra, y si se ha finalizado.

| Asignatura | Datos del alumno | Fecha |
|---------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Justificación del uso de la Tecnología Blockchain

En mi caso, el uso de la tecnología Blockchain es bastante importante ya que en un proceso de trazabilidad tener todos los procesos guardados en la Blockchain da al cliente una seguridad adicional de que los datos que se le están proporcionando son los correctos y realmente verificados y verídicos.

Ventajas:

Transparencia con el cliente.

Capacidad de comunicación.

Eficiencia de recursos ya que ahorra llamadas innecesarias entre el fabricante y el cliente.

Desventajas:

Privacidad.

Posibles errores en los Smart contracts.

Smart Contracts implementados:

Contrato **main.sol** que contiene todas las funcionalidades de añadir procesos, modificar estados, ver los diferentes procesos activos...

Contrato para generación de Procesos mediante los tokens no fungibles ERC721.

| Asignatura | Datos del alumno | Fecha |
|---------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Diagrama de Despliegue

Diagrama de despliegue – main

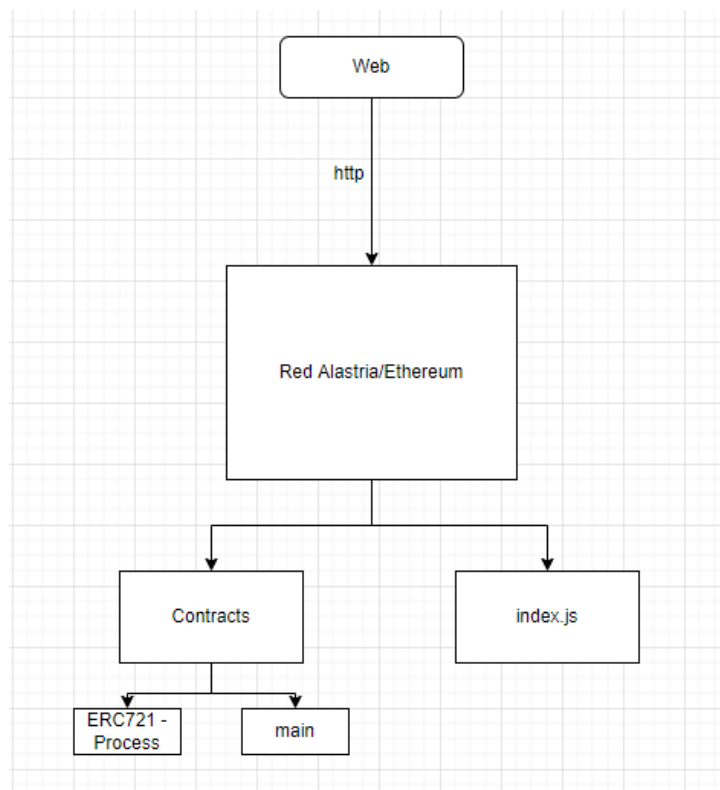
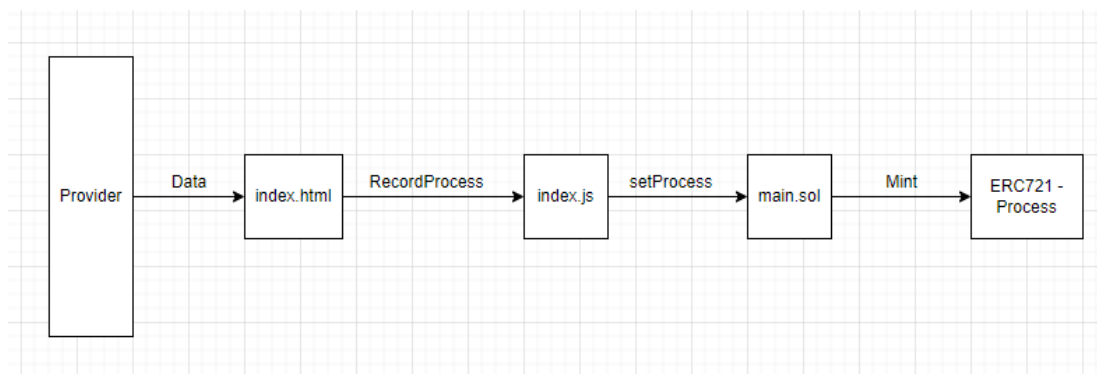


Diagrama ERC721 - Procesos



| Asignatura | Datos del alumno | Fecha |
|---------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Diagrama Frontend

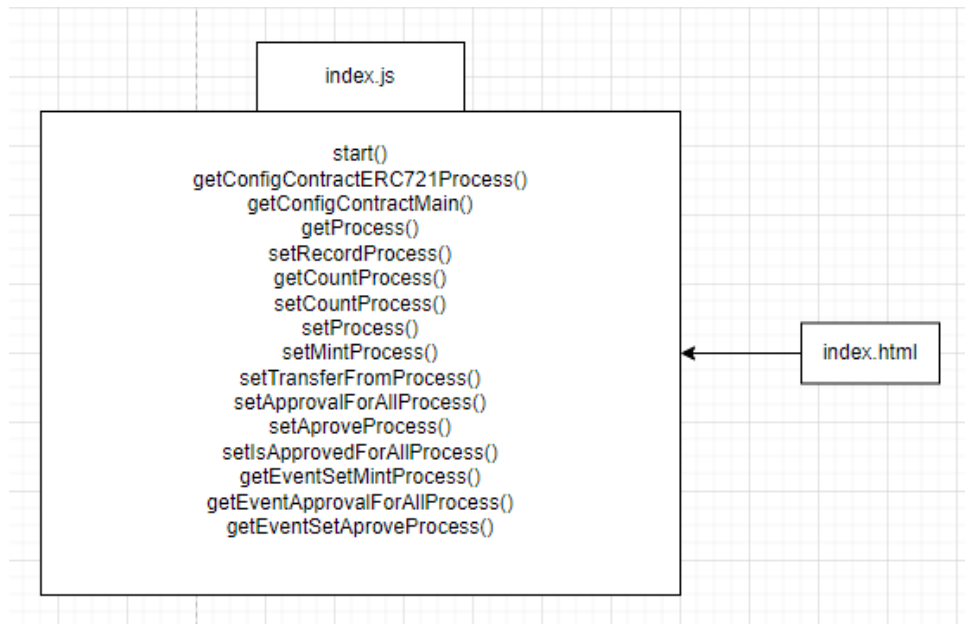
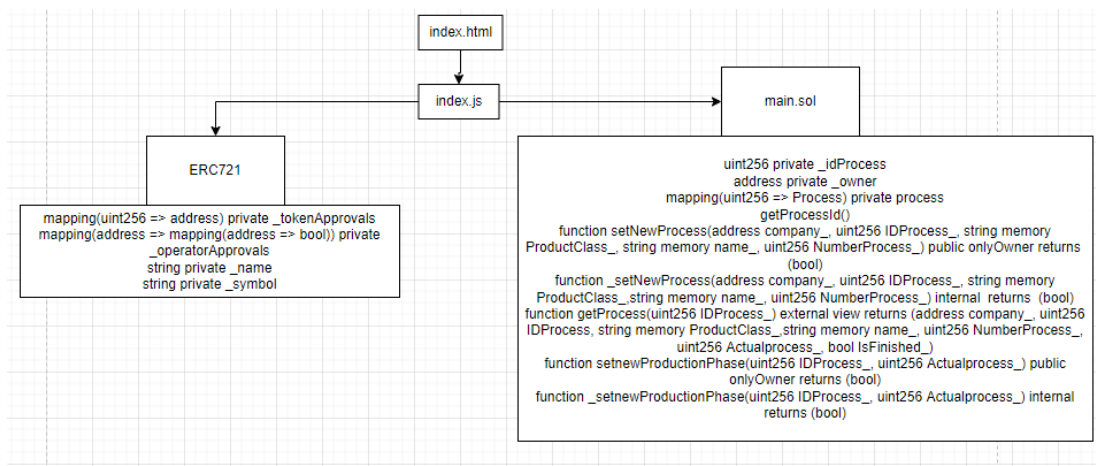


Diagrama Backend



| Asignatura | Datos del alumno | Fecha |
|---------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Instrucciones de Despliegue

Descargar el repositorio: `git clone https://github.com/alastria/test-environment.git`

```
marti@marti-VirtualBox:~$ git clone https://github.com/alastria/test-environment
.git
fatal: destination path 'test-environment' already exists and is not an empty directory.
marti@marti-VirtualBox:~$
```

Ejecutar los siguientes comandos de instalación:

```
cd test-environment/infrastructure/testnet/

git clone https://github.com/alastria/alastria-node.git
cd alastria-node/
git checkout develop
cd..
sudo -H $PWD/alastria-node/scripts/bootstrap.sh
rm -rf alastria-node
```

Comprovar la instalación:

```
go version
geth version
constellation-node -version
```

```
root@testenv:/home/ubuntu/test-environment/infrastructure/testnet# go version
go version go1.6.2 linux/amd64
root@testenv:/home/ubuntu/test-environment/infrastructure/testnet# geth version
Geth
Version: 1.8.12-stable
Git Commit: f64dbb25e4eb533878cd98791e33ad1abb7e97de
Quorum Version: 2.2.0-0.Alastria_EthNetstats_IBFT
Architecture: amd64
Protocol Versions: [63 62]
Network Id: 1337
Go Version: go1.9.5
Operating System: linux
GOPATH=
GOROOT=/usr/local/go
root@testenv:/home/ubuntu/test-environment/infrastructure/testnet# constellation
-node --version
Constellation Node 0.3.2
```

| Asignatura | Datos del alumno | Fecha |
|---------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Instalar versión posterior a la 8.0 de node:

```
sudo apt-get remove nodejs
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash
sudo apt-get install -y nodejs
node --versión
```

```
root@testenv:/home/ubuntu/test-environment/infrastructure/testnet# sudo apt-get
install -y nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 13 not upgraded.
Need to get 14.1 MB of archives.
After this operation, 70.5 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_8.x xenial/main amd64 nodejs amd64 8.17.0-
1nodesource1 [14.1 MB]
Fetched 14.1 MB in 0s (17.5 MB/s)
Selecting previously unselected package nodejs.
(Reading database ... 219449 files and directories currently installed.)
Preparing to unpack .../nodejs_8.17.0-1nodesource1_amd64.deb ...
Unpacking nodejs (8.17.0-1nodesource1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up nodejs (8.17.0-1nodesource1) ...
root@testenv:/home/ubuntu/test-environment/infrastructure/testnet# node --versio
n
v8.17.0
```

Levantar la red Alastria:

```
sudo ./bin/start_network.sh clean 1 2
ps -a
```

```
[*] constellation node at 9002 is now up.
Verify if /home/ubuntu/test-environment/infrastructure/testnet/logs/ have new fi
les.
root@testenv:/home/ubuntu/test-environment/infrastructure/testnet# ps -a
  PID TTY          TIME CMD
 2668 pts/1    00:00:00 sudo
 2669 pts/1    00:00:00 su
 2670 pts/1    00:00:00 bash
 6126 pts/1    00:00:00 geth
 6134 pts/1    00:00:23 constellation-n
 6353 pts/1    00:00:00 geth
 6365 pts/1    00:00:20 constellation-n
 6577 pts/1    00:00:00 geth
 6588 pts/1    00:00:00 ps
```


| Asignatura | Datos del alumno | Fecha |
|---------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Instalar Truffle:

```
sudo npm install -g truffle@5.0.18
```

truffle versión

```

giving output above:
Keccak bindings compilation fail. Pure JS implementation will be used.
+ truffle@5.0.18
added 94 packages from 307 contributors in 23.154s
alastria@alastria-VirtualBox:~$ truffle version
Truffle v5.0.18 (core: 5.0.18)
Solidity v0.5.0 (solc-js)
Node v8.17.0
Web3.js v1.0.0-beta.37
alastria@alastria-VirtualBox:~$

```

Crear un proyecto en Truffle:

```

alastria@alastria-VirtualBox:~$ mkdir hello-world-smart-contracts
alastria@alastria-VirtualBox:~$ cd hello-world-smart-contracts
alastria@alastria-VirtualBox:~/hello-world-smart-contracts$ truffle init

✓ Preparing to download
✓ Downloading
✓ Cleaning up temporary files
✓ Setting up box

Unbox successful. Sweet!

Commands:

  Compile:      truffle compile
  Migrate:      truffle migrate
  Test contracts: truffle test

alastria@alastria-VirtualBox:~/hello-world-smart-contracts$ ls -ltr
total 20
drwxrwxr-x 2 alastria alastria 4096 may 28 11:18 contracts
-rw-rw-r-- 1 alastria alastria 4235 may 28 11:18 truffle-config.js
drwxrwxr-x 2 alastria alastria 4096 may 28 11:18 test
drwxrwxr-x 2 alastria alastria 4096 may 28 11:18 migrations
alastria@alastria-VirtualBox:~/hello-world-smart-contracts$

```

Una vez tenemos el directorio desde donde ejecutaremos los contratos con truffle, tenemos que añadir los Smart contracts para su ejecución.

| Asignatura | Datos del alumno | Fecha |
|---------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Seguidamente nos conectamos a uno de los nodos de la red local para utilizar la consola JavaScript mediante el siguiente comando:

```
sudo geth attach ipc:network/general1/geth.ipc
```

```
alastria@alastria-VirtualBox:~/test-environment/infrastructure/testnet$ sudo geth attach ipc:network/general1/geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/general1/v1.7.2-stable-94e1e31e/linux-amd64/go1.9.5
coinbase: 0xf61c5a534591d2da95c2dc191e7a6a290892dbb9
at block: 355 (Fri, 22 Feb 2019 19:05:42 CET)
datadir: /home/alastria/test-environment/infrastructure/testnet/network/general1
modules: admin:1.0 debug:1.0 eth:1.0 istanbul:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0
> █
```

A continuación, consultamos y desbloqueamos una dirección Ethereum en un nodo de la red local:

```
web3.eth.accounts
```

```
web3.personal.unlockAccount(web3.eth.accounts[0], "Password", 0);
```

Configuramos la conexión con la red local:

```
development: {
  host: "127.0.0.1",
  port: 22001,
  network_id: "*",
  gas: 0xffffffff,
  gasPrice: 0x0,
  from: "<dirección extraída del nodo>"
}
```

A continuación, compilamos los contratos y los desplegamos:

```
sudo truffle compile
```

```
sudo truffle migrate
```

Ejecutamos el comando siguiente si queremos interactuar con la consola: `truffle console`

Finalmente hacemos el despliegue de los Smart contracts en el nodo de la UNIR:

```
sudo geth attach http://10.141.8.11:8545
```

```
web3.eth.accounts
```

Se muestran todas las cuentas disponibles.

En última instancia nos vamos en Remix compilamos los Smart contracts y, desde allí compilamos y hacemos el despliegue de los contratos en el proveedor de web3 en el endpoint de la UNIR: <http://10.141.8.11:8545>.

| Asignatura | Datos del alumno | Fecha |
|---------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Manual de Usuario

Registrar Nuevo Proceso de Producción

En este apartado lo que haremos será entrar todas las características del proceso que queremos añadir y rellenaremos los siguientes datos:

Company: nombre de la empresa que fabrica el producto.

IDProcess: cada proceso tiene un identificador único.

Product Class: tipo de producto según su categoría.

Name: nombre del producto.

Number of States: número de fases que hay durante la producción del proceso.

Register new Process

Company

IDProcess

Product Class

Name

Number of States

Register Process

| Asignatura | Datos del alumno | Fecha |
|---------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Modificar fase de producción del producto

En este formulario modificaremos el estado en el que está el producto que se está produciendo para que el cliente esté actualizado en cada momento del estado del producto.

IDProcess: identificador único del proceso de producción.

Process Phase: fase del producto de producción.

Set Next Production Phase

IDProcess

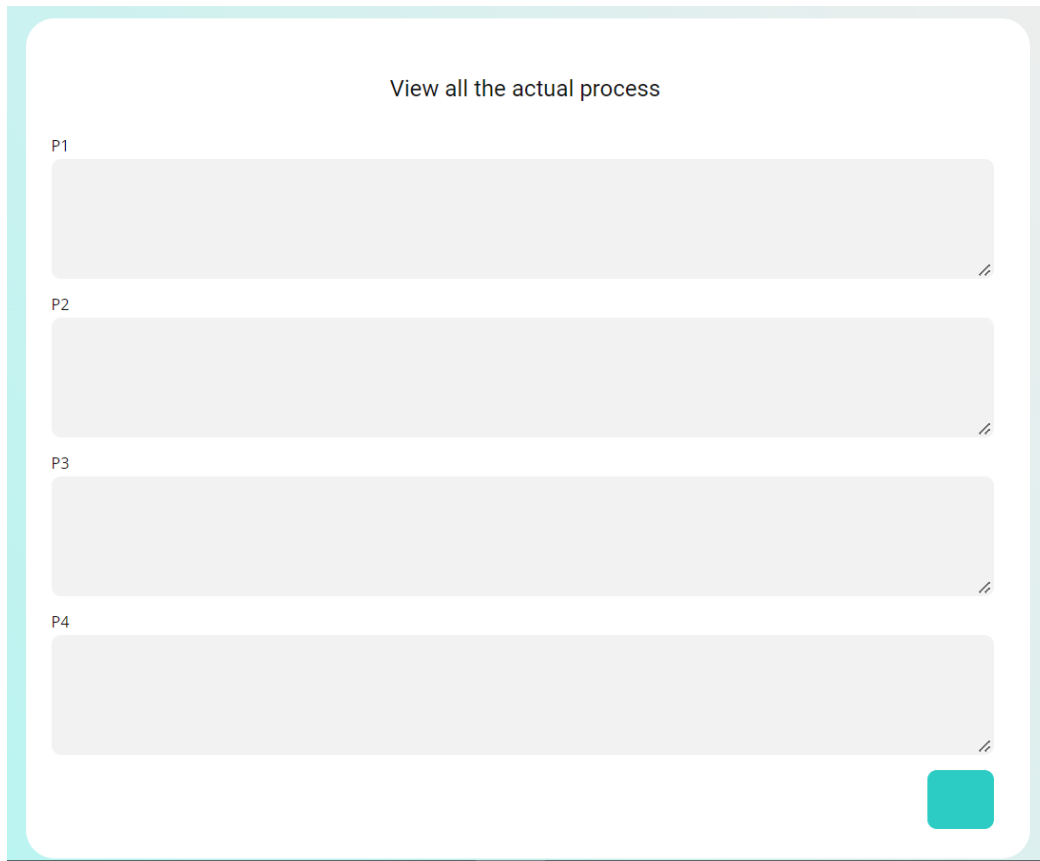
Process Phase

Set Next Production Phase

| Asignatura | Datos del alumno | Fecha |
|---------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Visualizador de los Procesos de Producción

En esta página el cliente podrá ver los diferentes productos que ha encargado en la fase en que están y si están terminados.



Pruebas – Main

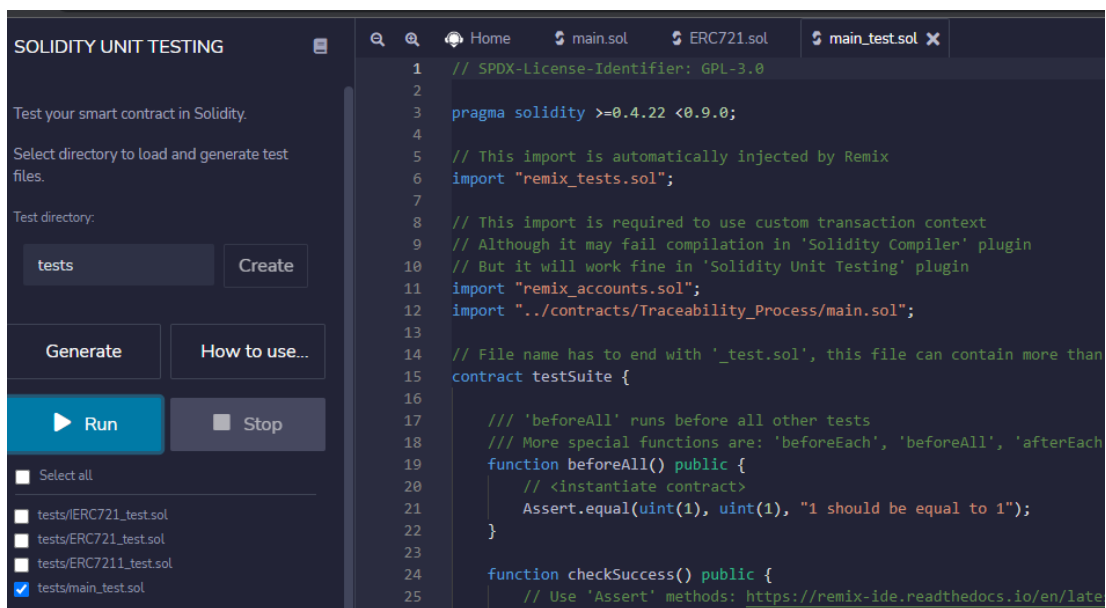
OpenZepelin ERC721:

El contrato ERC721 está basado en el proyecto OpenZepelin, por lo tanto, los test ya están hechos en su repositorio de código: <https://github.com/OpenZeppelin/openzeppelin-contracts/tree/master/test/token/ERC721>

| Asignatura | Datos del alumno | Fecha |
|---------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Pruebas Solidity Unit Testing:

He utilizado diferentes métodos para hacer el testing de los contratos, en este caso el **main.sol**. Dentro de remix tenemos el **plugin** de **Solidity Unit Testing** que permite hacer un test de los contratos generados:



Pruebas Truffle Testing:

Otra forma de hacer un testing de los contratos es mediante el truffle test que tiene incorporado el propio compilador truffle. Para hacer este test se ha instalado truffle, se han creado las carpetas correspondientes y finalmente se ha ejecutado el **truffle test**. Para poder hacer este test se ha tenido que crear un fichero **.js** e indicarle los mediante código **Javascript** los tests que queremos que pase. En los archivos adjuntados hay la carpeta "test" con los tests que se ejecutan dentro del fichero **test.js**.

Compiling your contracts...

=====

- > Compiling .\contracts\Address.sol
- > Compiling .\contracts\Context.sol
- > Compiling .\contracts\ERC165.sol
- > Compiling .\contracts\ERC721.sol

| Asignatura | Datos del alumno | Fecha |
|----------------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

```
> Compiling .\contracts\IERC165.sol
> Compiling .\contracts\IERC721.sol
> Compiling .\contracts\IERC721Metadata.sol
> Compiling .\contracts\IERC721Receiver.sol
> Compiling .\contracts\Migrations.sol
> Compiling .\contracts\Strings.sol
> Compiling .\contracts\main.sol
> Compiling .\contracts\mapping.sol
> Compiling .\contracts\mapping_process.sol
> Compiling .\test\main.sol
> Compiling .\test\mapping.sol
> Compiling .\test\mapping_process.sol
> Compilation warnings encountered:
```

Warning: Unreachable code.

```
--> project:/contracts/mapping_process.sol:10:29:
```

```
|
10 |     for (uint i=0; i<4; i++){
    |                ^^^
```

,Warning: Unreachable code.

```
--> project:/test/mapping_process.sol:10:29:
```

```
|
10 |     for (uint i=0; i<4; i++){
    |                ^^^
```

,Warning: Unused function parameter. Remove or comment out the variable name to silence this warning.

```
--> project:/contracts/mapping_process.sol:8:23:
```

```
|
8 |     function indexOf( mapping(uint256 => InfoProcess.Process) storage Process, string memory
productclass, uint2 ...
    |                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```


| Asignatura | Datos del alumno | Fecha |
|----------------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

--> project:/test/mapping_process.sol:8:154:

```
|
8 | ... berprocess, uint256 actualprocess, string memory n, bool finished) internal view retu ...
|               ^^^^^^^^^^^^^^^^^^^^^
```

,Warning: Unused function parameter. Remove or comment out the variable name to silence this warning.

--> project:/test/mapping_process.sol:8:171:

```
|
8 | ... 56 actualprocess, string memory n, bool finished) internal view returns (uint256 ID ...
|               ^^^^^^^^^^^^^^^^^^^^^
```

,Warning: Function state mutability can be restricted to pure

--> project:/contracts/mapping_process.sol:8:5:

```
|
8 | function indexOf( mapping(uint256 = ... view returns (uint256 IDProcess_) {
|   ^ (Relevant source part starts here and spans across multiple lines).
```

,Warning: Function state mutability can be restricted to pure

--> project:/test/mapping_process.sol:8:5:

```
|
8 | function indexOf( mapping(uint256 = ... view returns (uint256 IDProcess_) {
|   ^ (Relevant source part starts here and spans across multiple lines).
```

| Asignatura | Datos del alumno | Fecha |
|---------------|------------------------|------------|
| Trabajo Final | Apellidos: Prat Mayans | 04/07/2022 |
| | Nombre: Martí | |

Conclusiones

Una vez finalizado el TFE creo que he interiorizado todos los conceptos explicados durante el Máster ya que he podido lanzar el proyecto exitosamente pudiendo realizar el TFE de forma exitosa utilizando los contratos ERC721 y, implementando toda la parte **frontend** con **html**, **css** y **Javascript** de lo que me siento muy satisfecho.

Finalmente, el despliegue y las pruebas del proyecto lo he hecho con **Ganache** ya que con la red Telsius de Alastria no funcionaba correctamente.