MEMORIA

Diagrama de despliegue de la solución

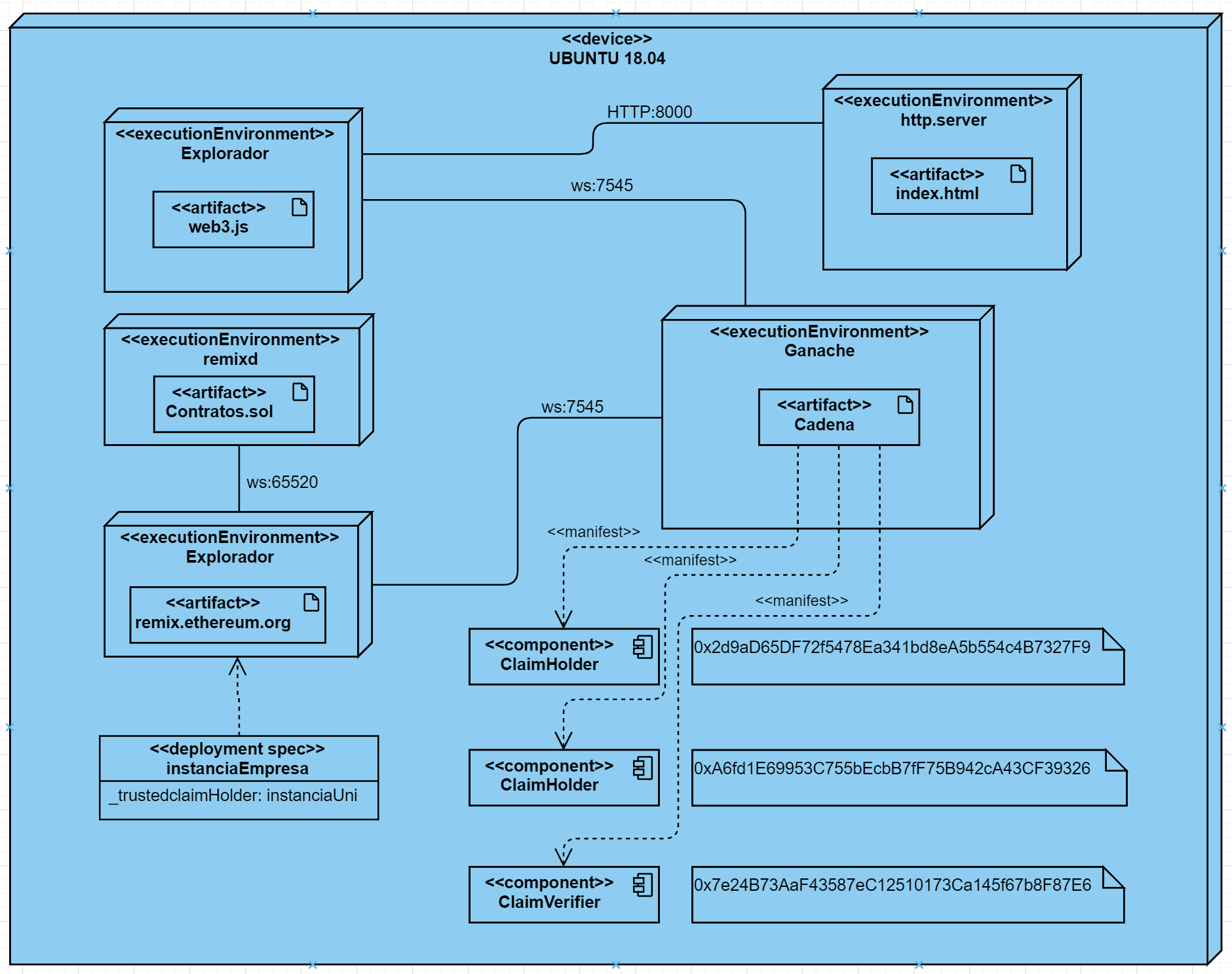
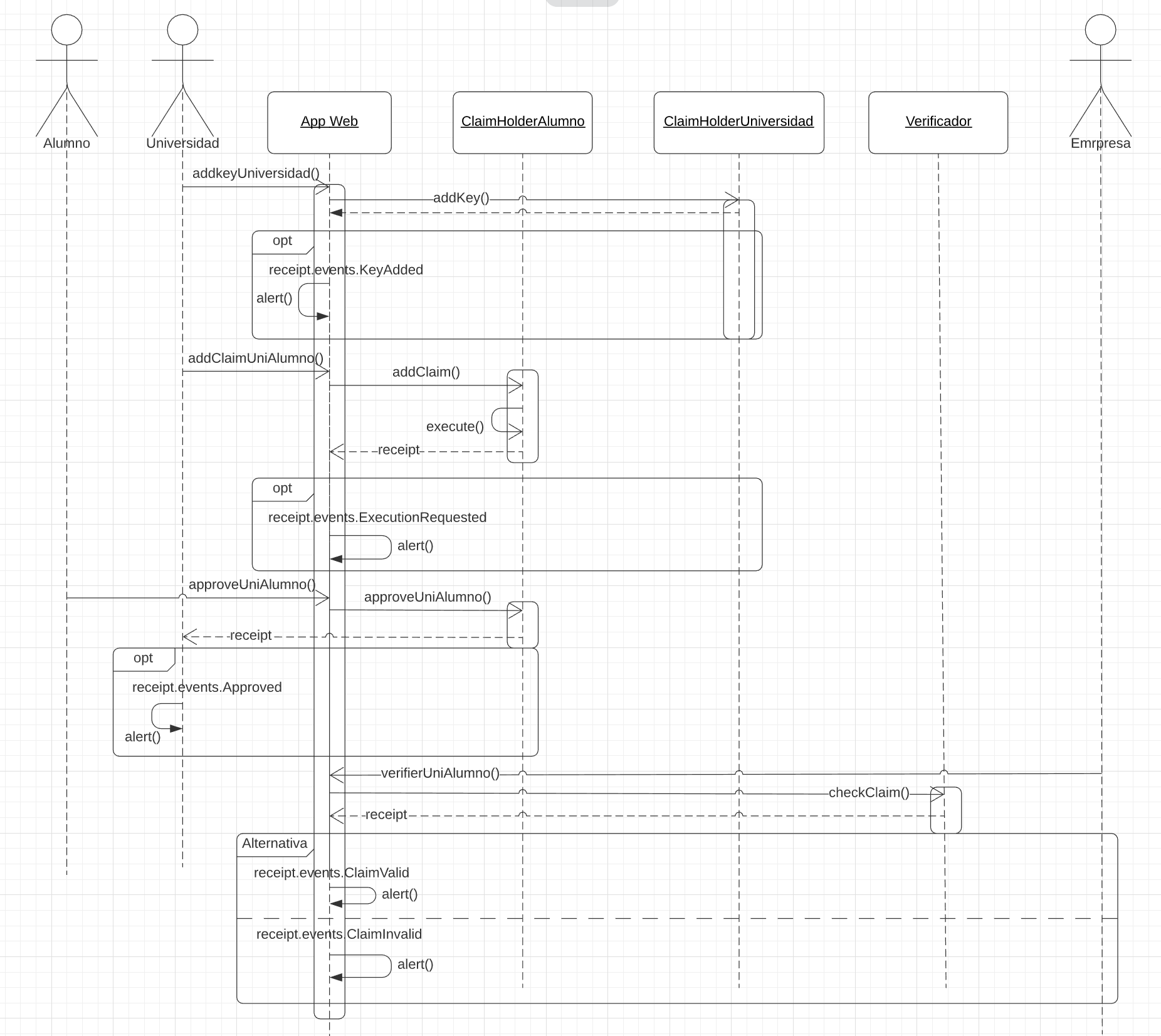


Diagrama de secuencia que muestra el flujo completo de la aplicación: emisión, aprobación y verificación



Instrucciones de despliegue del sistema

Para desplegar el sistema propuesto, primero se instalan los programas necesarios:

* ganache
* remixd
* python

Python viene instalado por defecto.

Primero vamos a iniciar una cadena de blockchain en ganache con quickstart.

Ahora tenemos un servidor de cadena de bloques al que accedemos en la siguiente dirección:

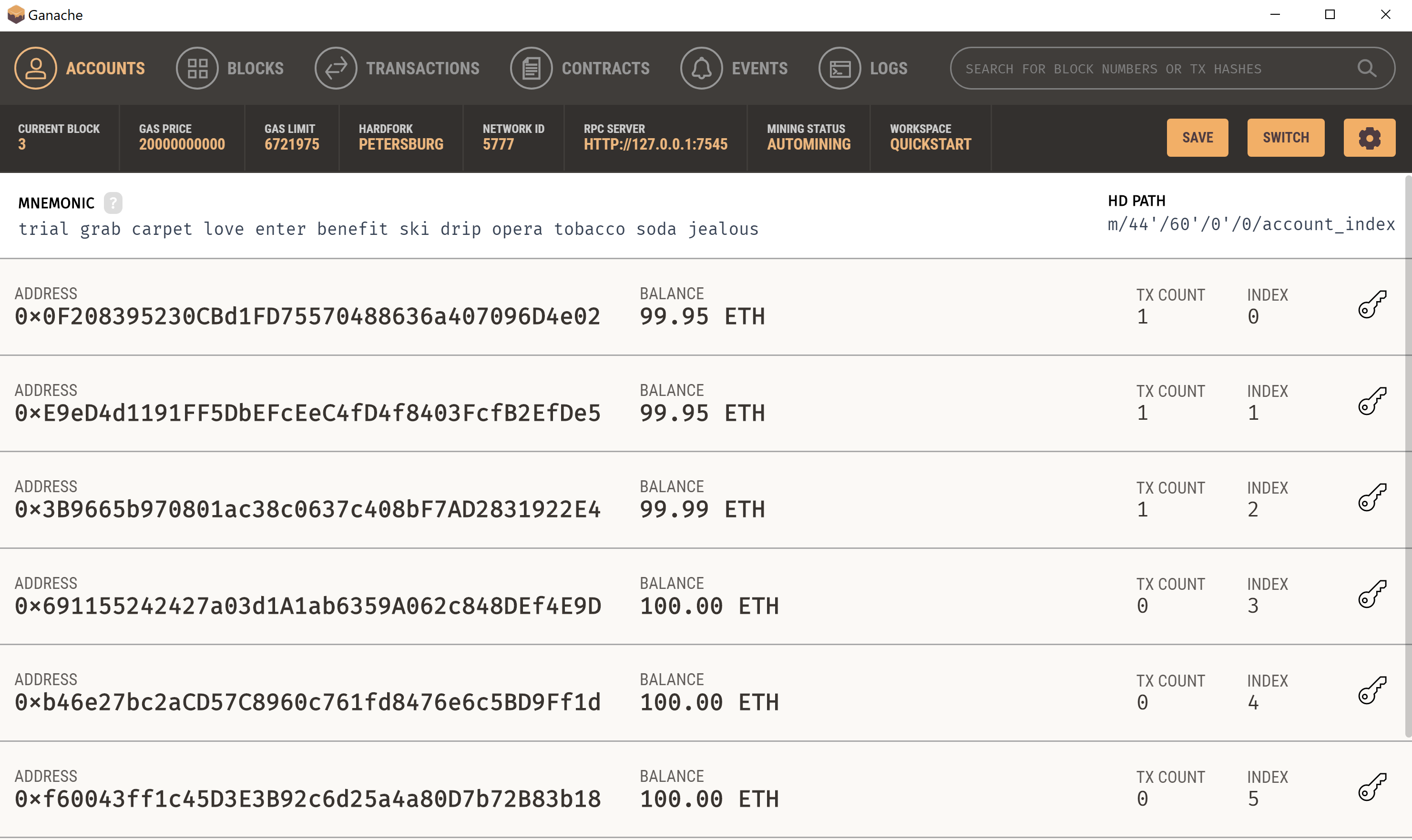
RPC SERVER = <HTTP://127.0.0.1:7545>

Ganache nos crea un conjunto de cuentas disponibles que estarán accesibles desde remix.

La mnemonic usada para crear las cuentas en este proyecto es:

“trial grab carpet love enter benefit ski drip opera tobacco soda jealous”

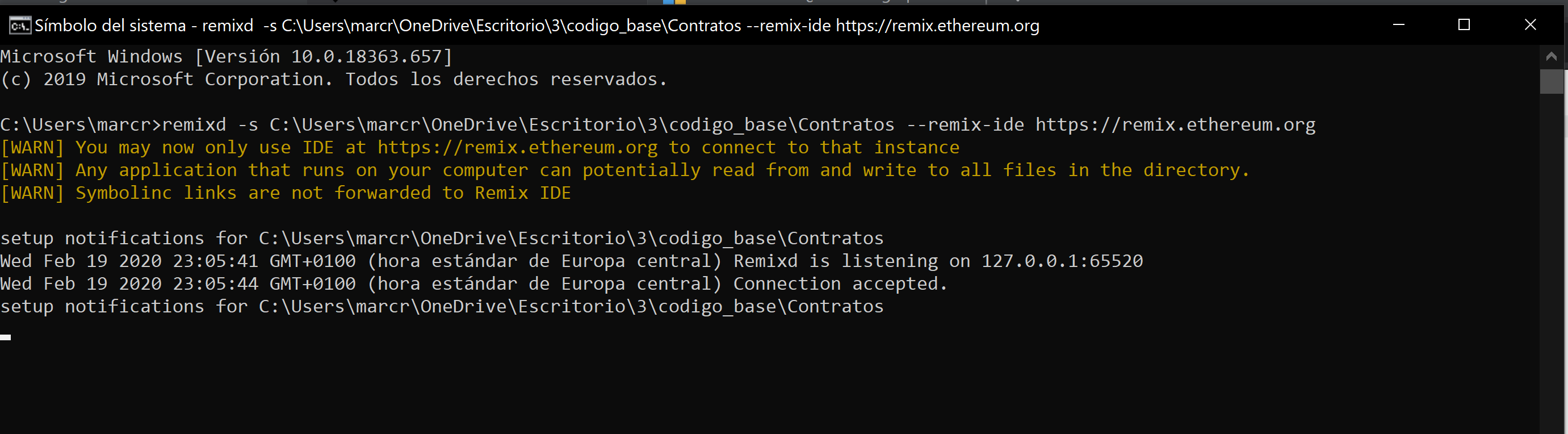
Usando esta configuración obtendremos las mismas direcciones.



Ahora vamos a desplegar los smartcontracts mediante remix. Primero vamos a conectar nuestro servidor de remixd para que la web pueda cargar los contratos.

Vamos a especificar la ubicación de la carpeta donde se encuentran los contratos:

remixd -s <shared folder> --remix-ide https://remix.ethereum.org



Accedemos a:

<https://remix.ethereum.org>

Conectamos la web de remix con el server de remixd:

Clicamos Connect to Localhost en el apartado File de la pestaña Home.

Connection will start a session between https://remix.ethereum.org/ and your local file system *ws://127.0.0.1:65520* so please make sure your system is secured enough (port 65520 neither opened nor forwarded).

Los contratos ubicados en esa carpeta aparecen ahora en la web de remix.

Ahora vamos a compilar y desplegar los contratos en la cadena de ganache.

Hacemos click encima de los contratos ClaimHolder y Claimverifier y los compilamos desde solidity compiler con la versión 0.4.22.

Para desplegarlos hay que conectar la web de remix con ganache.

Desde la pestaña “Deploy & run transactions” seleccionamos como “environment” web3 provider. Nos pide el endpoint y vamos a pasarle la dirección y el puerto donde corre ganache:

<http://localhost:7545>

Cada contrato se despliega desde una dirección distinta: 

La account[0] despliega un ClaimHolder - cuentaAlumno

La account[1] despliega un ClaimHolder - cuentaUniversidad

La account[2] despliega un ClaimVerifier, ahí vamos a especificar el constructor \_trustedClaimHolder que es la dirección del contrato ClaimHolder desplegado desde la cuenta de la Universidad - cuentaEmpresa.

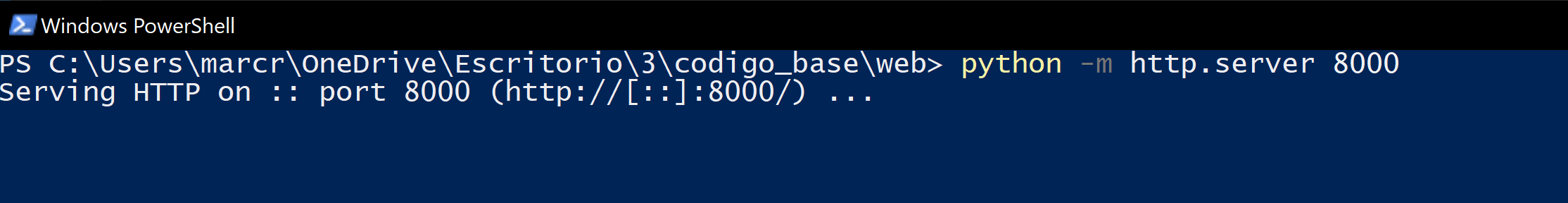
Clicamos Deploy para cada contrato con su cuenta correspondiente y vemos la transacción ejecutada tanto en la consola de remix como en ganache.



Para poder interactuar con nuestros contratos desde la aplicación web es necesario establecer un servidor web.

Desde la carpeta donde se aloja nuestro archivo index.html vamos a iniciar un servidor Python mediante el comando en terminal:

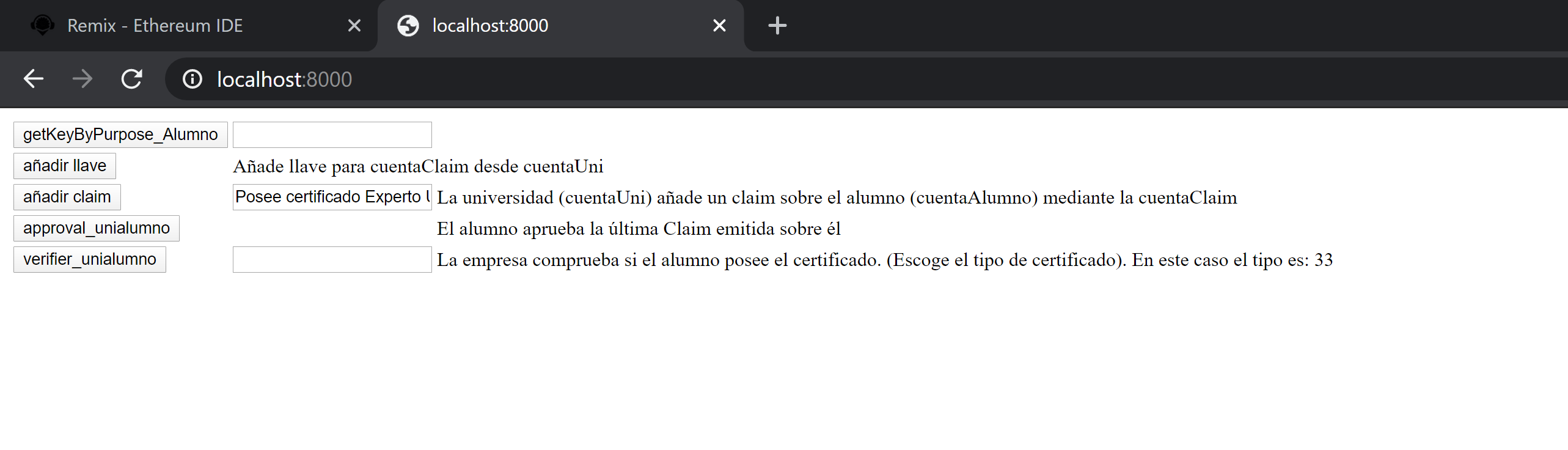
python -m http.server 8000



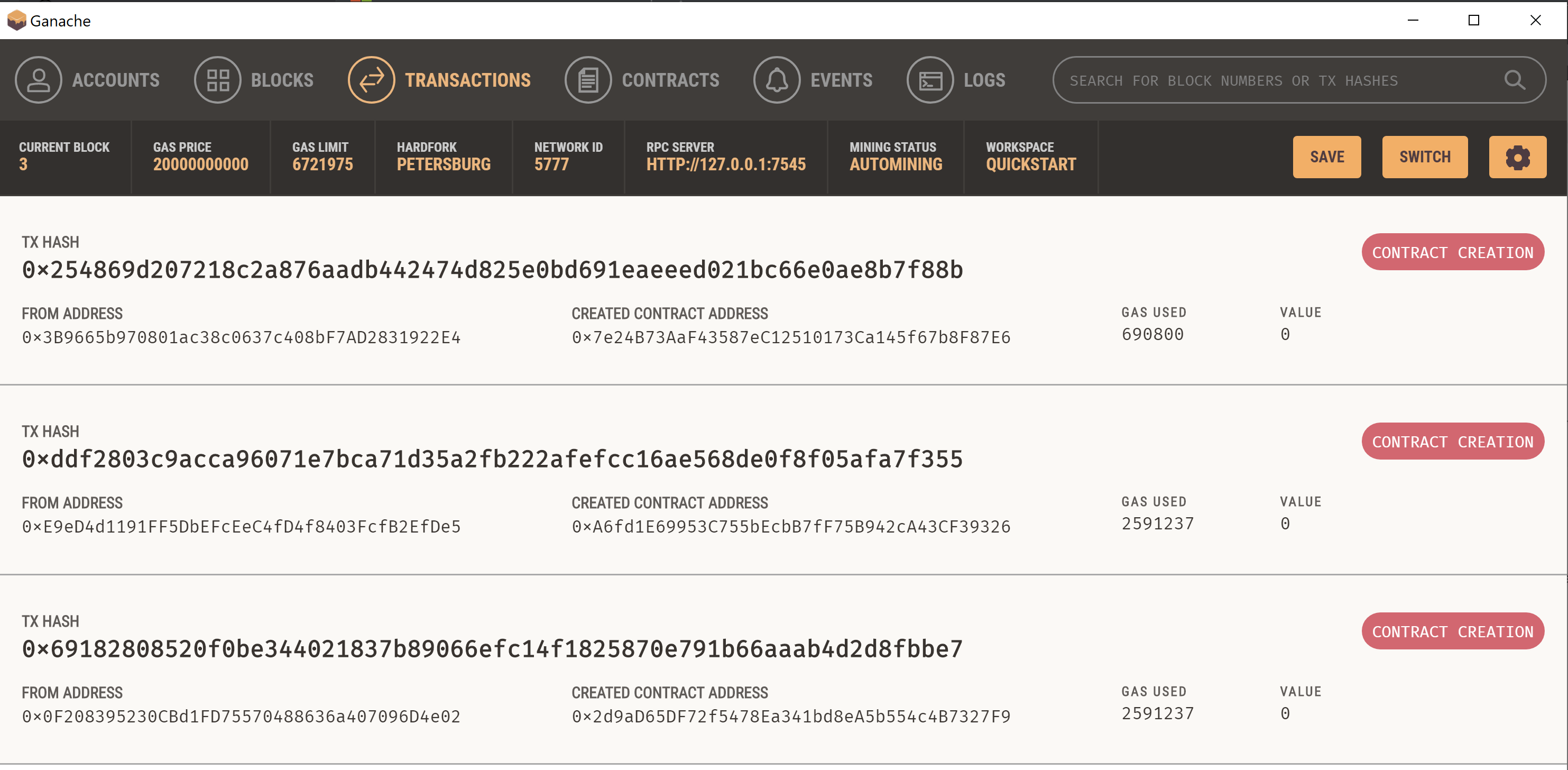
Nos muestra el siguiente mensaje conforme el servidor está funcionando:

Serving HTTP on :: port 8000 (http://[::]:8000/) ...

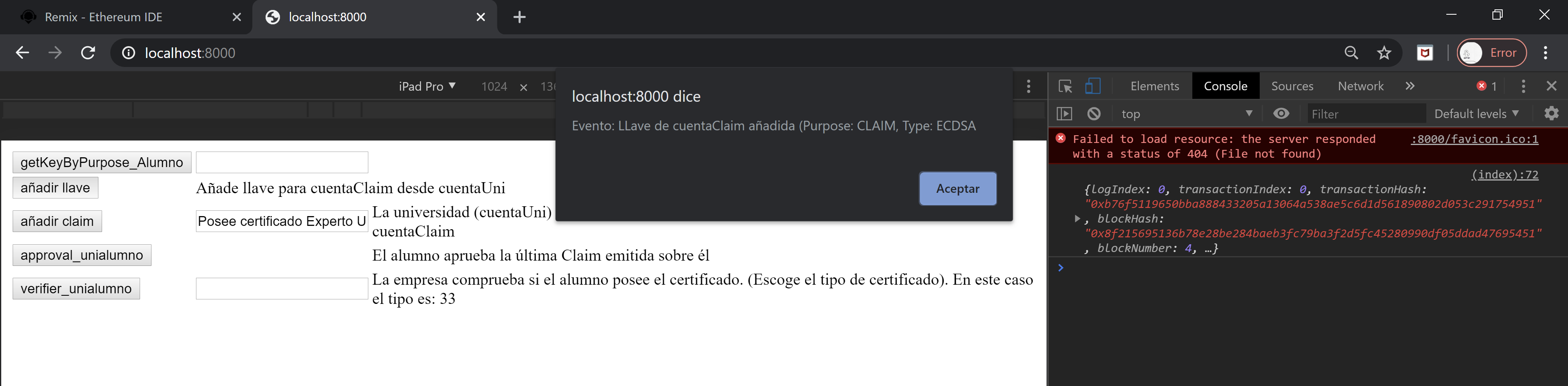
Si ahora accedemos desde el navegador a la dirección <http://localhost:8000> nos va a mostrar el contenido de nuestro archivo index.html.



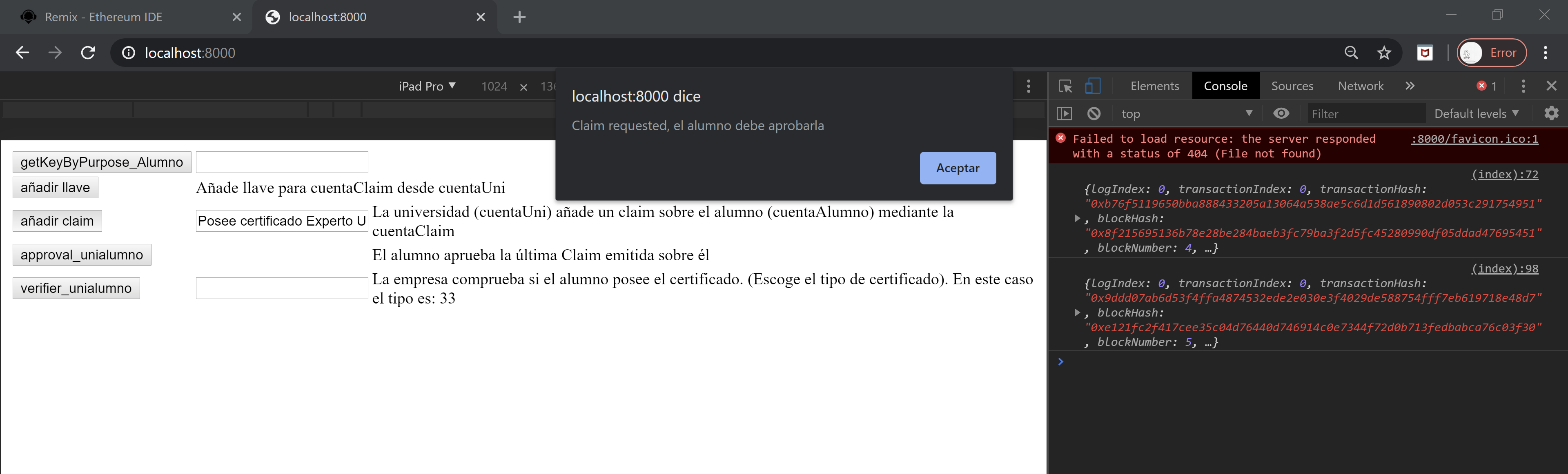
Manual de usuario, incluyendo capturas de pantalla de ejemplo, mostrando el uso del sistema



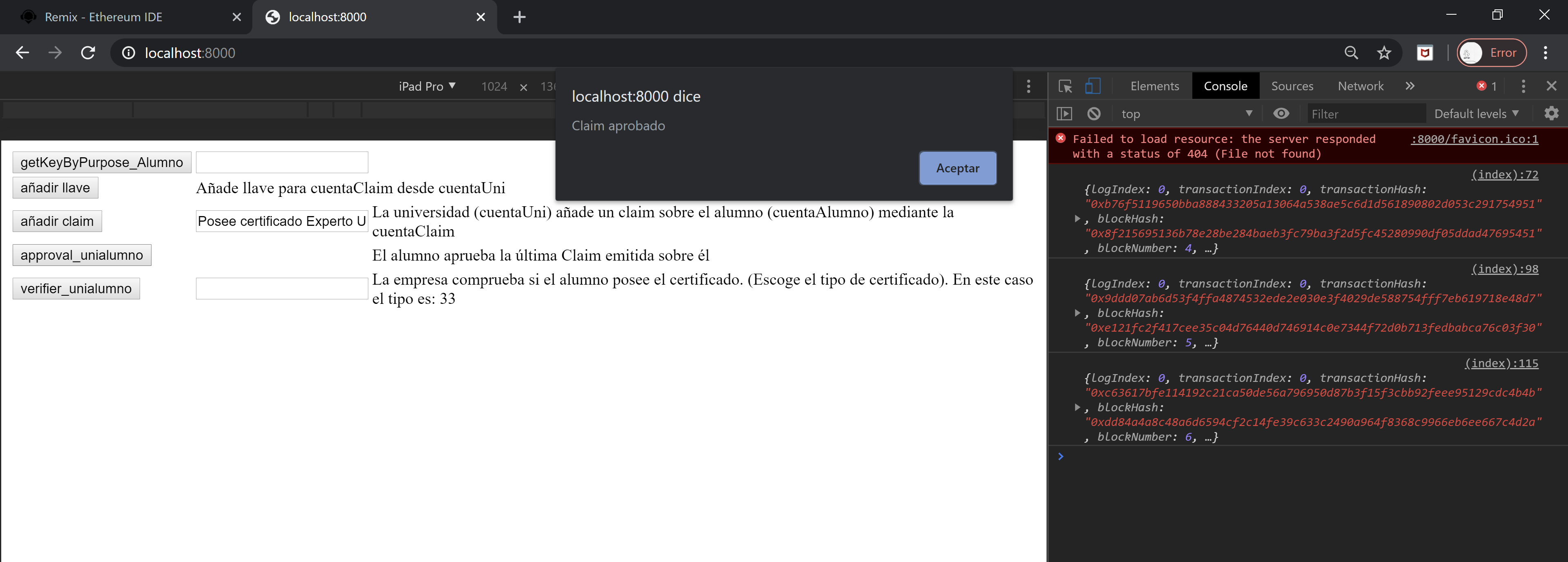
1. Click en añadir llave devuelve evento KeyAdded.



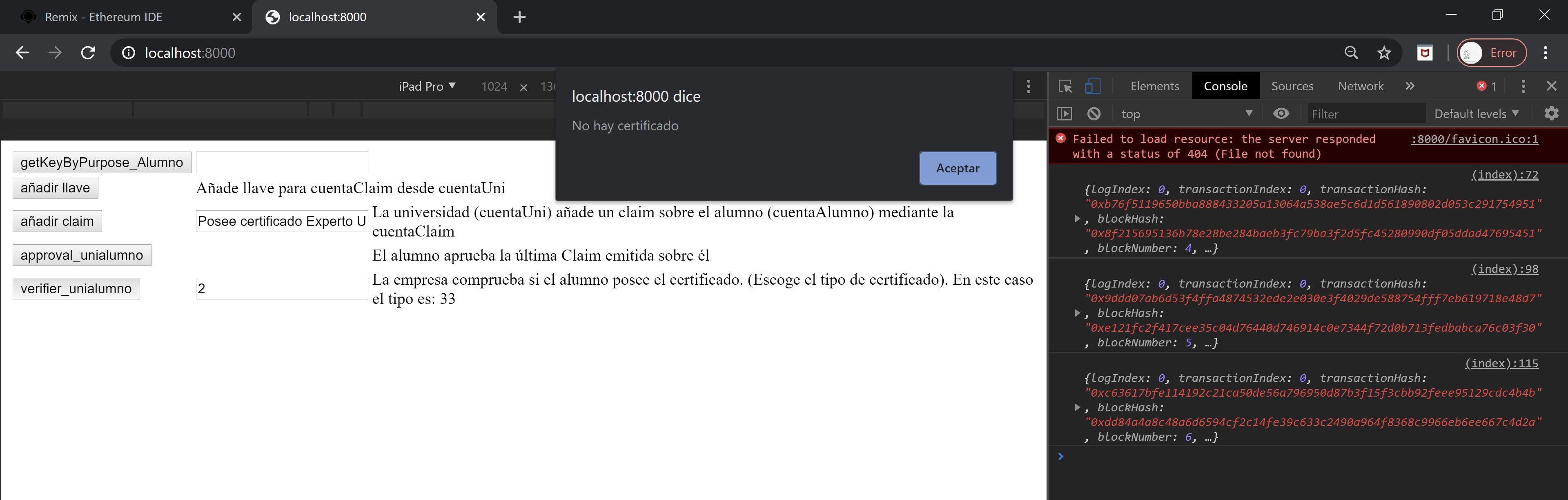
1. Click en añadir claim devuelve evento ClaimAdded.



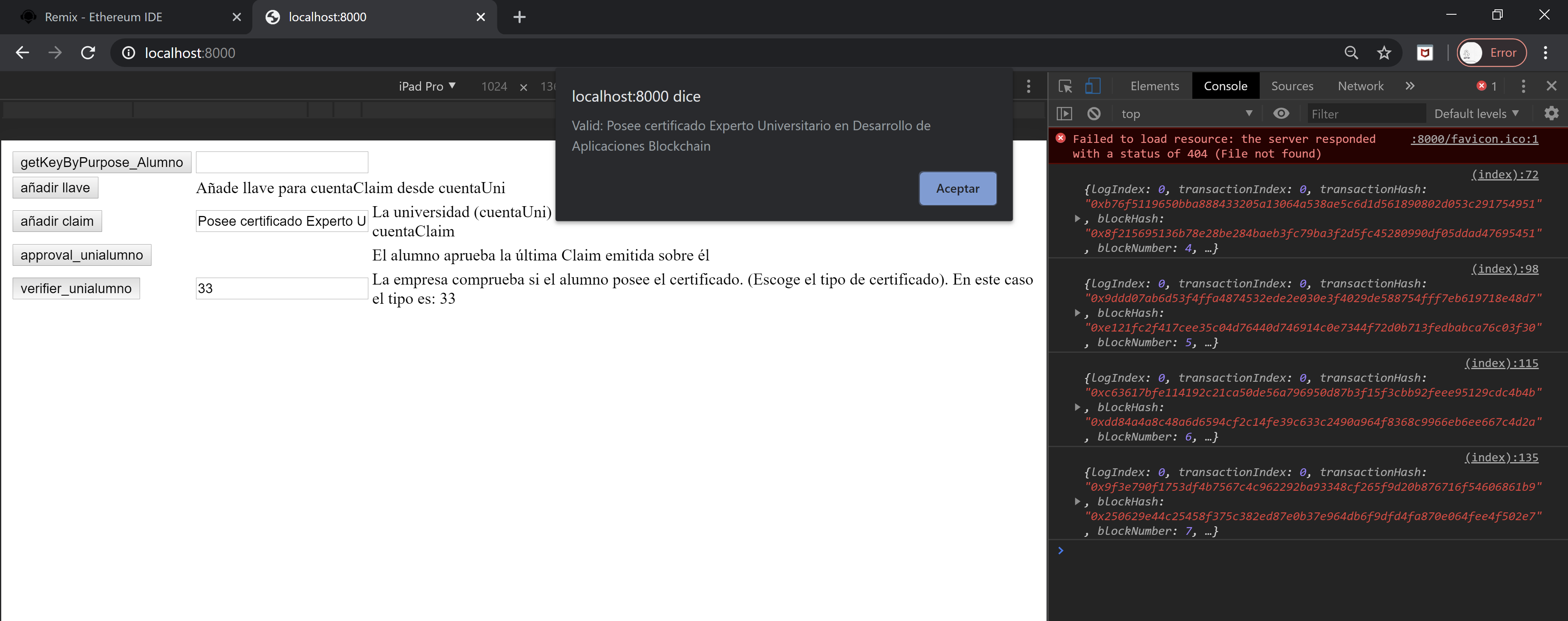
1. En el evento anterior hemos solicitado que se añada una claim, pero el alumno debe aprobarla con click en approval\_unialumno. Devuelve el evento approved.



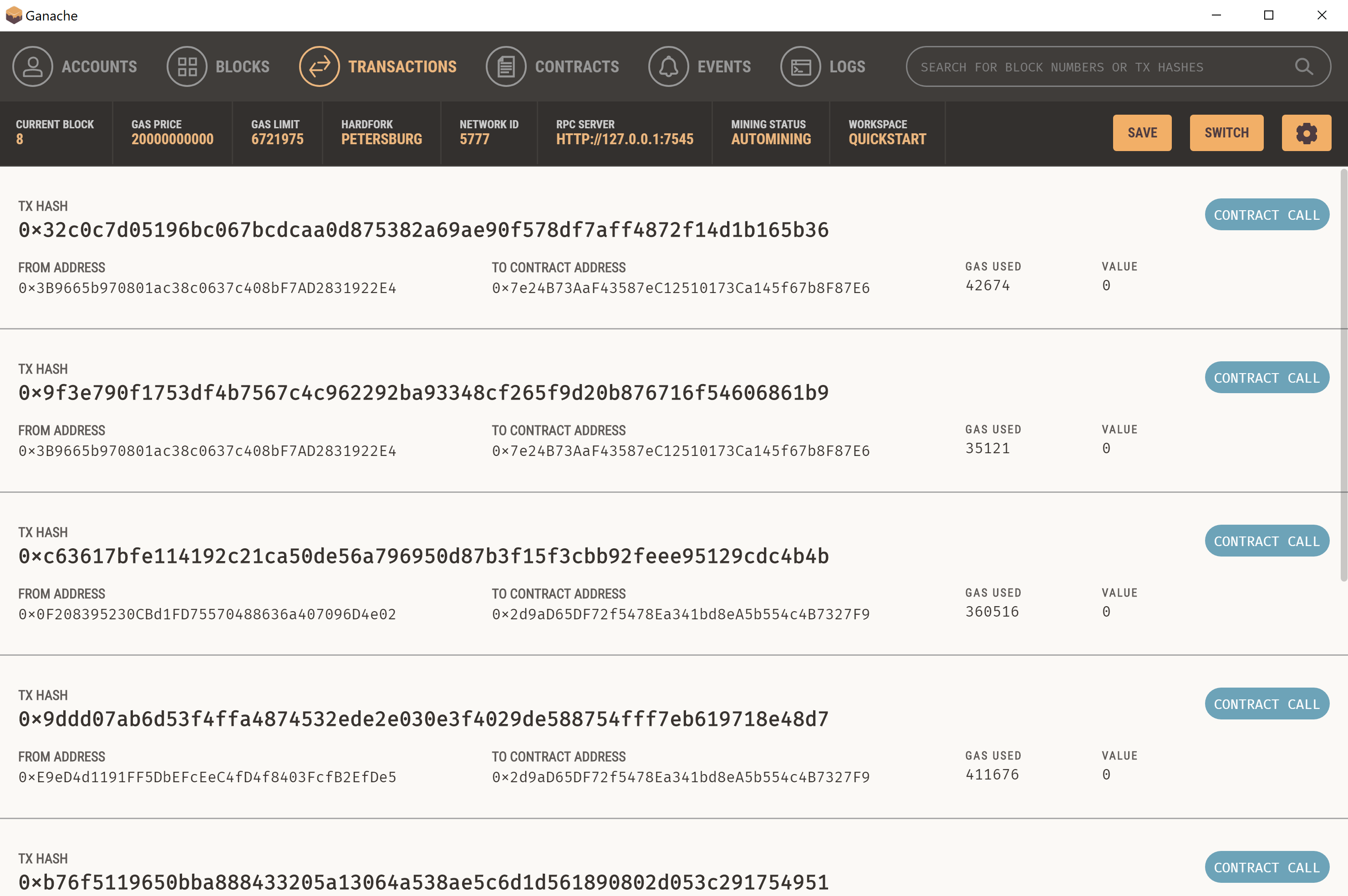
1. La empresa verifica si el alumno posee el certificado. Probamos con tipo de certificado 2. Al no ser el certificado correspondiente nos devuelve el evento ClaimInvalid.



1. El tipo de certificado que tiene el alumno es 33. Al introducir el tipo correcto nos devuelve el evento ClaimValid.



Se muestran las transacciones en ganache:



Se muestran las transacciones en remix:

