

## ETHEREUM TEAM PROJECT

### Overview

Se trata de desarrollar el software para crear a través de una web desarrollada en REACT los servicios siguientes:

1. Crear redes ETH usando el software geth (go ethereum).
2. Poder añadir nodos y quitar nodos
3. Poder borrar la red entera compuesta por varios nodos.
4. El protocolo de consenso será PoA (proof of authority)
5. Todos los nodos se crearán en la misma máquina.
6. Se debe de poder acceder desde metamask instalado en la misma máquina donde corres los nodos.
7. Hacer un explorer que nos dé el balance de una cuenta, la información de una tx y de un bloque.
8. Hacerlo tanto en windows como el ubuntu y en mac si es posible.

### Repositorio base

Este es el repositorio base a partir del cual hay que hacer el proyecto.  
<https://github.com/jviejocodecripto/proyecto-base.git>

### Sobre las redes Ethereum

Las redes Ethereum tiene los siguientes parámetros:

- Un chainID (1 es la mainnet)
- Un port http
- Un port para unirse a nodos de la misma red
- Un ipcpath (para comunicarse via memoria)

Para crear una red y un nodo usaremos la lógica siguiente para no tener problemas con los puertos:

```
NUMERO_NETWORK = 1  
NUMERO_NODO = 1
```

Y se calculan los puertos

```
El directorio donde estarán los datos será: const NETWORK_DIR =  
`eth${NUMERO_NETWORK}`
```

Y DENTRO DEL DIRECTORIO ESTARAN LOS DATOS DE CADA NODO

eth1/nodo1 (directorio para el nodo1 de la red1)  
eth5/nodo4 (directorio para el nodo4 de la red 5)

```
const HTTP_PORT = 9545 + NUMERO_NODO + NUMERO_NETWORK * 20
const AUTHRPC_PORT = 9553 + NUMERO_NODO + NUMERO_NETWORK * 20
const NODO = `nodo${NUMERO_NODO}`
const PORT = 30404 + NUMERO_NODO + NUMERO_NETWORK * 20
const IPCPATH = `\\\\.\\pipe\\${NETWORK_CHAINID}\\${NODO}.ipc`
const DIR_NODE = `${NETWORK_DIR}/${NODO}`
```

En el formulario en el que creemos la red solo pediremos numero de red y numero de nodo.

Como cuenta para precargar fondos poner alguna vuestra de metamask, para que tengáis fondos.

## Backlog

1. Start:
  - a. Task crear el proyecto en github para tener todos el mismo código y poder sincronizarlo. Crear un proyecto con dos carpetas: front-end y back-end a partir del proyecto base <https://github.com/jviejocodecripto/proyecto-base.git>
  - b. Task, estudiar el proyecto base.
  - c. Ver los videos de <https://loom.com/share/folder/9a57f94fd8254f359058319cac0957ea>
2. Aplicación web (front end)
  - a. Task: Debe tener una **cabecera** con el logo y nombre del servicio. El nombre del servicio será Build Private Ethereum Networks. Debe tener un **footer** que tendrá los links: Quienes somos, privacidad, términos y condiciones e icono de redes sociales.
  - b. Task: La aplicación tendrá una barra con la siguiente opción
    - i. **Networks**
  - c. Task: El links nos saca la **lista de redes** debe tener un botón que permite **añadir una red nueva**. Se debe también de poder **borrar una red** y también visualizar la red
  - d. Task: Cuando visito una red me tiene que decir los nodos que tiene y debo tener la posibilidad de añadir un nuevo nodo, o borrar uno existente.
  - e. Task: Cuando visualizo la red debe de poder obtener la **lista de los bloques**
  - f. Task: Dado un **bloque** debo de poder ver las transacciones
  - g. Task: Data una **transacción** debo de poder ver los datos de esta

# GUÍA DE ESTUDIO

- h. Task: Debo poder implementar un **faucet** para cargar monedas en mi billetera metamask.
  - i. Task: Hacer un **traspaso de fondos** de una cuenta del metamask a otra en un formulario.
  - j. Task: Dada un address debo de poder ver su **balance**
3. Aplicación node js (backend)
- Vais a usar un repositorio inicial, que tiene la lógica de crear un nodo de Ethereum de manera programática.
- a. Task crear la aplicación nodejs con express, add el cors.
  - b. Task ruta crear la network (ver el fichero **genesis** para ver qué datos necesitamos). Creamos en nodo número 1
    - i. Implementar ruta crear network usando el software **geth**.
    - ii. Hacer lo mismo con **docker** container. (opcional)
    - iii. Hacer lo mismo con **kubernetes**. (opcional)
  - c. Task ruta **add** un nodo a una network
  - d. Task ruta **delete** a nodo a una network
  - e. Task ruta **faucet** para enviar monedas a una cuenta
  - f. Task ruta nodo, para ver un nodo
  - g. Task ruta balance para ver el **balance de una cuenta**
4. Hacer tareas de documentación
- a. Task Video presentación del producto
  - b. Task Documentar el README.md
5. Software a usar: (<https://geth.ethereum.org/downloads/>)