

Implementación Sistema de Votación

1. Diseño

1.1. Main

- Definir localidades (localidades.txt)
- Generar votantes
- Levantar nodos ETH localmente (*)
- Definir nodo ETH para cada centro de votación (*)
- Generar y levantar centros de votación
- Configurar escenario electoral
- Iniciar proceso de votación

1.2. Generador de Votantes

Crear v votantes:

- Crear clave privada y pública.
- Asignar nombre, correo, saldo inicial de 10.000.000 wei, localidad electoral, centro de votación dentro de localidad, clave privada y pública.
- Asociar una dirección a la clave pública.
- Seleccionar 1% para ser candidatos a los cargos(asamblea/congreso) por localidad.

1.3. Generador de Votos

Crear e inicializar tantos hilos como indique el nivel de concurrencia de envío

Cada hilo envía un voto:

- Escoger votante
- Escoger aleatoriamente entre los candidatos de su localidad
- Enviar voto a su centro de votación

1.4. Escenario Electoral (Elecciones Legislativas asamblea/congreso)

Envía la información del escenario al **centro1** para:

- Registrar votantes
- Registrar candidatos
- Registrar localidades

1.5. Centro de votación

Crear n centros de votación:

- Asignar puerto
- Asociar a su localidad
- Crear e iniciar hilo que se encargará de comunicarse con la red
- Recibe intenciones de voto
- Envía la intención de voto (transacción) al nodo eth (*)

1.6. Contrato inteligente

- Inicializar escenario electoral
 - Registrar localidades

- Registrar votantes
- Registrar candidatos
- Reinicializar escenario si el contrato ya está creado
- Recibir intención de voto
- Cerrar proceso de votación
- Reportar ganadores con % votos para cada candidato y abstención
- Reporte por localidades

1.7. Visualizador

Resumen:

-

Reporte detallado:

-

2. Consideraciones

Se realizaron las siguientes implementaciones para una implementación y testeo más rápida y eficiente:

- El archivo ***data/localidades.txt*** contiene el conjunto de localidades electorales, junto con su cantidad de votantes y centros de votación.
- El archivo ***data/centros.txt*** contiene los centros de votación con sus respectivos puertos
- Las propuestas para votar por representante de asamblea y congreso, se representan con 1 y 2 respectivamente.
- El votante vota por al menos un representante a la asamblea y uno para el congreso
- (*) Un sólo nodo Ethereum local usando por default brownie, con geth se pueden crear varios nodos locales y conectarlos, pero al crear la red para usarla con brownie no usa la configuración especificada para la red, por lo que solo ejecuta geth sin ningún argumento y empieza a sincronizar la red completa de Ethereum. Según lo que investigue, ganache-cli (por defecto usado por brownie) no permite más de un nodo.

3. Bugs

- La concurrencia no está bien implementada, ocurren votos dobles y el programa no termina
- Si se prueba con muchos votantes en total (más de 100) puede dar el error: ***ConnectionResetError: [Errno 104] Connection reset by peer***

4. Ejecución y prueba

4.1. Ejecutar ***brownie run scripts/VotingSystem.py***

Es el archivo principal para realizar las pruebas, puede crear los votantes, candidatos y centros de votación para luego iniciar el proceso electoral