Universidad Simón Bolívar Departamento de Computación y Tecnología de la Información
CI-4852 - Tópicos Especiales en Computación: Tecnologías Blockchain
Septiembre 2021 - Enero 2022
12-11250 Natascha Gamboa

Sistema de Votación con contrato inteligente

Caso de uso: Votaciones de representante de asamblea y gobernador por localidad

Nombre del contrato: Voting

El siguiente contrato tiene como objetivo poder realizar un proceso de votación, la estructura es la siguiente:

- Estructuras
  - Localidad
    - nombre
    - cantidad de votantes
    - votos ejercidos
  - Votante
    - nombre
    - indicador para verificar si ya votó
    - localidad en la que vota (esto permite llevar el conteo de votos de la localidad)
  - Candidato
    - nombre
    - votos recibidos
    - Cargo al que se postula
- Variables
  - Conteo de votos
  - Estado en el que se encuentra el contrato (Creado, Votando, Cerrado). Esto sirve para poder restringir las funciones y que no se pueda votar si el proceso ya se cerró.
  - Registro de localidades
  - Registro de votantes válidos
  - Registro de candidatos ganadores por localidad
  - Mapeo desde una localidad para obtener sus candidatos. (Esto permite poder sumar los votos para los candidatos)
- Funciones
  - Constructor del contrato que actualiza el estado a creado e inicializa el conteo de votos
  - Reseteo del proceso de votación que vuelve a inicializar el estado a creado, el conteo de votos y los ganadores
  - Dar comienzo al proceso de votación
  - Registrar localidad
  - Registrar votante
  - Registrar candidato
  - Ejercer voto
  - Cerrar proceso de votación y realizar el conteo de votos para escoger los ganadores por localidad
  - Indicar el porcentaje de votos que recibió cada ganador y de abstinencia
  - Indicar la cantidad de votos que recibió cada ganador y de abstinencia

● 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 se investigó, ganache-cli (por defecto usado por brownie) no permite más de un nodo.

● Bugs

- El porcentaje de votos que recibió cada ganador no se calcula correctamente
- Si se prueba con muchos votantes en total (más de 100) puede dar el error: ***ConnectionResetError: [Errno 104] Connection reset by peer***

● Ejecución y prueba

Se necesita instalar brownie para poder probar el contrato inteligente, se recomienda instalar la versión estable

Para configurar el entorno de prueba se deben modificar los archivos:

- ***scripts/data/localidades.txt*** para indicar los nombres de la localidades junto con la cantidad de votantes y centros de votación
- ***scripts/data/centros.txt*** para indicar la cantidad de centros de votación, el nombre y puerto de cada uno
- ***scripts/VotingSystem.py*** se modifica para crear la cantidad de centros de votación necesarios (suma de los centros de votación por localidad)

Finalmente para ejecutar y probar el contrato se utiliza el siguiente comando dentro de la carpeta raíz del proyecto

- ***brownie run scripts/VotingSystem.py***