Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI-4852 - Tópicos Especiales en Computación: Tecnologías Blockchain
Septiembre-Diciembre 2021

Proyecto 2 (30 %)

En este proyecto Ud. deberá diseñar e implementar, para la red Ethereum, un conjunto de Contratos Inteligentes para realizar un proceso electoral. Adicionalmente debe realizar validaciones empíricas y teóricas de su implementación.

Objetivo: Al finalizar el proyecto, el estudiante entenderá el proceso de desarrollo de Contratos Inteligentes basado en el uso de prácticas para reducir los riesgos de seguridad.

Nota: Cada equipo tendrá un escenario electoral diferente. Este enunciado establece las pautas generales del proyecto. Cada grupo debe establecer las condiciones específicas para su escenario.

1. Generadores

En esta sección Ud. desarrollará los generadores que permitan crear los escenarios para la correcta ejecución de su aplicación.

1.1. Generador de Votantes

Debe desarrollar una utilidad que permita crear un conjunto de votantes ficticios. Cada uno de ellos tendrá asociado un nombre, correo, localidad electoral, una clave privada, una pública y una dirección asociada a esta última. Cada elector tendrá un saldo inicial de 10.000.000 wei.

Requerimientos:

- Debe definir un conjunto de localidades electorales.
- A cada votante debe asignársele en forma aleatoria un centro de votación en su localidad.

- De entre los votantes creados, se debe seleccionar un subconjunto razonable de éstos (1 %) para usarlos como candidatos a los diferentes cargos a ser elegidos. Esta actividad depende del escenario de votación que le corresponda a su grupo.

La ejecución será de la siguiente forma:

```
Sintaxis: genVotante -f <archivo-localidades>
```

Donde el archivo de localidades tiene el siguiente formato:

```
Local1  numVotantes1 numeroCentrosVotación1
...
Localn  numVotantesn numeroCentrosVotaciónn
```

donde:

- **Locali**: Representa el nombre de la localidad, es decir, la entidad política. La localidad representará un Estado (en el caso de Venezuela). No será necesario manejar una subdivisión de distritos, municipios etc.
- **numVotantesi**: Número de votantes inscritos para esa localidad.
- **numeroCentrosVotacióni**: Número de centros de votación disponibles en esa localidad. Estrictamente hablando, todos los votantes pueden ejercer su derecho desde sus casas. Sin embargo, imagine los votantes requieren ir a un centro por garantías de seguridad etc.

1.2. Generador de Votos

Este generador tiene como función generar y enviar los votos a los centros de votación.

Requerimientos:

- El voto debe ser generado aleatoriamente entre las opciones disponibles en base al escenario implementado.
- Cada voto debe ser enviado al centro de votación correspondiente.
- El envío de los votos debe ser concurrente.

- Debe generar votos para cada localidad manteniendo el porcentaje de abstención entre el 10 % y el 30 %. Estos valores deben ser de fácil modificación en su código.

La ejecución de este generador será la siguiente:

```
Sintaxis: genVotos -n <archivo centro> -nc <n>
```

Donde:

- <n>: es el nivel de concurrencia de envío de votos.
- <archivo centro>: Ubicación del servidor del centro de votación: nombre y puerto.

```
n
centro1 puerto1
centro2 puerto2
...
centron puerton
```

1.3. Escenario electoral

Este utilitario será el responsable de interactuar con el `centro1` para realizar el registro del escenario de votación:

- Registrar Votantes
- Registrar Candidatos
- Registrar Localidades

Para más detalle ver las actividades siguientes.

2. Centros de Votación

Debe escribir un servidor sencillo que reciba en forma concurrente la intención de voto de los votantes, cree las transacciones correspondientes (el campo *from* debe corresponder

al votante)[1] y las envíe a un nodo de la red Ethereum[2].

La ejecución de este servidor:

```
Sintaxis: centroVotacion -n <nombreCentro> -p <puerto> -f <archivo datos nodo>
```

Donde:

- <nombreCentro>: Nombre asignado a este centro de votación.
- <puerto>: puerto para recibir peticiones del generador de votos.
- <archivo datos nodo>: Contiene los datos de la ubicación del nodo Ethereum al cual se le enviarán las transacciones.

Nota: El `centro1` recibirá la información de los escenarios del proceso electoral. En otras palabras, el proceso de ejercer los votos no puede comenzar antes de la configuración del escenarios.

3. Software para Votación: Contrato Inteligente

Diseñar Contratos Inteligentes que permitan llevar a cabo las elecciones asociadas al escenario de votación asignado.

Su contrato debe:

- **Inicializar el escenario**: Debe crear el contrato e inicializar las estructuras necesarias.
- **Reinicializar el escenario**: Si el contrato ya está creado, se debe re-inicializar las estructuras necesarias.
- **Registrar las localidades**: Recibir las localidades que participan en la elección. Éstas definen las zonas electorales. En caso de elecciones presidenciales, la zona corresponde a la unión de las localidades.
- **Registrar los votantes**: Registra y asocia cada votante con su localidad.

[1] Si lo considera procedente, puede generar las transacciones en el generador de votos
[2] Deberá levantar un conjunto de nodos Ethereum en forma local.

- **Registrar a los candidatos**: Registra cada candidato en la localidad especificada. Éstos deben ser votantes válidos.
- **Recibir la intención de voto de los votantes**: Debe realizar las verificaciones correspondientes al escenario asignado.
- **Cerrar el proceso de votación**: Finaliza el proceso. Luego de esto, no se pueden recibir más votos.
- **Reportar los ganadores con sus porcentajes**: Una vez finalizado el proceso, retorna un resumen de los resultados obtenidos. Incluyendo la abstención.
- **Dar un reporte por localidades**: Una vez finalizado el proceso, retornar un pequeño informe dando los resultados parciales (incluyendo la abstención) para todas las localidades.

Nota: Los detalles específicos dependen del escenario. Por ejemplo, el registro de localidades y votantes es diferente en escenarios de elecciones del Poder Legislativo o del Poder Ejecutivo, ya sea nacional o local.

4. Visualizador

Este utilitario mostrará tanto el resumen como el reporte detallado del proceso electoral.

Como actividad extra: Estudie la posibilidad de presentar una interfaz web que le permita a los votantes ejercer su derecho al voto y obtener los resultados del proceso una vez finalizado.

5. Escenarios

Los escenarios posibles comprenden al menos los siguientes:

- **Elecciones con una sola vuelta**: Incluye elección del Presidente del país y de un Gobernador por localidad. Al finalizar el proceso, el nuevo presidente/gobernador es el candidato con mayor número de votos en su zona electoral.
- **Elecciones con segunda vuelta**: Incluye elección de Presidente y de Gobernadores. Al finalizar la primera vuelta, los dos candidatos con mayor número de votos en su

zona electoral, pasarán a un segundo proceso de votación (segunda vuelta) con todos los votantes. Al finalizar la segunda vuelta, el nuevo Presidente/Gobernador será el que tenga el mayor número de votos en cada zona electoral.

- **Elecciones presidenciales de dos niveles**: Similar el esquema electoral usado en los Estados Unidos. En la primera fase los votantes eligen a los representantes de la localidad y en la segunda fase, estos representantes eligen quién será el nuevo presidente.
- **Elecciones legislativas**: Se seleccionan los 2 candidatos con mayor votos por cada localidad a ser representantes de esa localidad ante la asamblea/congreso.

En el caso del Gobernador, la zona electoral es la localidad y en caso del Presidente es la unión de las localidades.

Para la asignación del escenario, envíe un correo a su profesor notificando su grupo de trabajo.

6. Informe

Su informe debe mostrar el diseño e implementación de su aplicación. Debe mostrar que sus contratos cumplen con las prácticas y patrones de seguridad estudiados. Y que no presenta los anti-patrones que deben ser evitados.

7. Sugerencias

- Haga una lista de requerimientos y funcionalidades
- Clasifique y ordene los requerimientos.
- Mantenga contacto constante con su compañero. Vayan integrando y probando los avances individuales.
- Informe oportunamente a su profesor sobre problemas encontrados.