Trabajo Práctico Especial

Programación de Objetos Distribuidos

Sistema de votación

Grupo 3:

Esteban Kramer - 55200

Oliver Balfour - 55177

Martina Scomazzon - 55410

Bianca Ritorto - 57082

Introducción

El objetivo del trabajo práctico es realizar un sistema remoto thread-safe para realizar una elección, contabilizando votos, definiendo a los ganadores y permitiendo la fiscalización de los mismos.

Decisiones de diseño o implementación de los servicios

Se decidió que una vez que las elecciones hayan finalizado, no se pueden volver a abrir. De esta forma, el servicio solo funciona con otra tanda de elecciones si se reinicia su ejecución.

Dado que implementamos el método propuesto por la cátedra STV donde se expresa la siguiente fórmula:

cantidad de votos a transferir = (votos de la segunda elección / total de votos del ganador) * surplus

Donde surpuls = exceso de votos del candidato ganador, resulta que no siempre dará un resultado entero por lo tanto se toma el piso de la cuenta, resultando en la posibilidad de la pérdida de un voto. Consideramos que esto no debería afectar mucho en la decisión final de los votos ya que se asume que será una gran cantidad de votos total.

Si hay un partido que tiene 0 votos, se lo incluye de todas formas en el .csv con el porcentaje en 0 ya que se decidió que será más útil e informativo tener la lista completa de candidatos y sus porcentajes, antes que dejar afuera a aquellos que no obtuvieron votos.

En el sistema de votos AV, si el candidato tenía solo dos elecciones y por alguna razón su primer elección tuvo que ser eliminada, y nuevamente su elección restante debe ser eliminada, se elimina y luego el voto se pierde ya que no tiene más opciones.

En algunas partes del código, por razones de simplicidad de lectura y mejor comprensión, se optó por dejar de lado aspectos relacionados con el uso de Java 8 como el uso de lambdas cómo métodos anónimos en algunas partes del código.

Criterios aplicados para el trabajo concurrente

El sistema consiste en un servidor y cuatro clientes para la administración, fiscalización, votación y consultas. Para obtener una comunicación asincrónica se definió en el servidor una interfaz remota de callback con la que se notifica a los clientes los cambios.

De forma que, para que los objetos sean remotos se instancian y luego exportan con *UnicastRemoteObject*.

En algunas partes del código se utiliza la Paralelización de los streams provista por java que podría no llegar a ser la más óptima.

Potenciales puntos de mejora y/o expansión

El trabajo podría extenderse para que se implementen otros métodos de votación, como por ejemplo los cardinales donde se califica a los candidatos en vez de asignarles una posición en un ranking. Ejemplos de esto son Evaluative Voting (EV) y Star Voting (SV).

Otra posible mejora es realizar chequeos de validación. Por ejemplo, de formato y contenido para los archivos de entrada. Asimismo la aplicación tendría que poder ser extendible para que se ingresen nuevos candidatos y partidos políticos sin modificar el código cada vez que se quiere realizar esa modificación, y también tendría que poder modificarse la cantidad de ganadores o de cantidad de provincias siendo los mismos parametrizables.

Por último por cada servidor tendría que ser posible tener más de una elección y ser persistente sin ningún inconveniente.