

XYZ es la empresa que eligió la Registraduría de cierto país para sistematizar las elecciones de los distintos estamentos de elección popular. Para el próximo año, hay aproximadamente 50 candidatos y 100 millones de votantes en total, distribuidos en alrededor de 1000 ciudades del país. La empresa requiere diseñar un sistema confiable, seguro y altamente disponible para la jornada de las próximas y subsiguientes elecciones.

Toda la estructura de las agrupaciones de votantes por mesa, ciudad, etc. ya ha sido modelada en una base de datos en PostgreSQL, con los ciudadanos aptos para votar, debidamente registrados.

Por razones de seguridad, debe desarrollarse un sistema que use de base el estilo arquitectónico de capas, que permita realizar la consulta en la base de datos sin que los clientes accedan directamente a ella. En la capa más interna, estará la base de datos (BD). En la siguiente capa, estará el componente de consulta (servidor), que es el único que podrá acceder a la BD. Y por último, en la capa más externa, estarán los clientes, que sólo podrán hacer consultas a la capa intermedia (componente de consulta o servidor).

Para determinar si lo contrata a usted, ha concebido una prueba que consiste en desarrollar un sistema de software que debe considerar las anteriores restricciones y los siguientes requerimientos:

- E1. Los grupos de votantes por cada ciudad, se agrupan por número de cédula en mesas de votación. Las mesas están ubicadas en colegios y universidades, y para la agrupación se tiene en cuenta la dirección de residencia que el votante tendrá que haber registrado previamente en el sistema central.
- E2. A cualquier ciudadano se le debe permitir consultar en qué sitio debe votar, por una interfaz de usuario sencilla, ingresando el documento de identidad.
- E3. El sistema debe ser escalable, pero sobre todo, altamente disponible. Se deben poder añadir más clientes en tiempo real.
- E4. Todas las estaciones de consulta (clientes) deben comunicarse con el sistema de consulta (servidor) usando un protocolo de comunicación ICE.
- E5. El servidor debe poder atender concurrentemente a todos los clientes que hagan consultas sobre la mesa de votación, con una latencia máxima de 1 segundo, aunque esta cantidad de tiempo podrá variar.
- E6. Para efectos de simulación y prueba, cada cliente debe implementar un pool de N consultas para ejecutar concurrentemente, donde N lo podrá ingresar el usuario final al iniciar el programa cliente.
- E7. Entre el cliente y el servidor debe implementarse el patrón Observer. El servidor es el observado, y los clientes son los observadores que se registran en el primero.
- E8. El servidor, en su rutina principal, debe ejecutar un ciclo en el que pregunte al usuario que digite un mensaje. El usuario digitará una ruta que incluye el nombre de un archivo. Este archivo tendrá una lista de documentos de identidad (uno por línea) a los que se desea consultar la mesa de votación. Si la cantidad de líneas es menor a la cantidad de consultas que un computador puede realizar sin perder consultas ni abortarse, él mismo debe hacerlas. En caso contrario, debe distribuir secciones de la lista entre los clientes registrados en él, notificándoles como sus Observadores, para que cada cliente realice las consultas de los documentos de identidad que le correspondió, sobre el servidor, en el menor tiempo posible (el tiempo estimado deseable es máximo un segundo).
- E9. El servidor debe calcular el número de factores primos del documento de identidad y hacer la consulta en la BD la mesa de votación. Si el número de factores primos es primo, debe retornar
- E10. El sistema debe ser auditable. En particular, en los clientes debe generarse un log con columnas separadas por coma. Las columnas son: la CC de consulta, y para cada CC, la respectivo mesa de votación, un indicador de si el número de factores primos es primo (1 si lo es, 0 si no lo es), y el tiempo que se demoró el sistema en responder, etc., que permita verificar la correctitud de las respuestas. En el servidor, debe dejarse la misma información por cada línea, pero antecedido del id del cliente que lo consultó, en un solo archivo de salida. En la última línea del archivo de salida, el número total de consultas realizadas y el tiempo total de ejecución.

Para el enunciado anterior, desarrollar y entregar:

- A) Análisis de requerimientos funcionales.
- B) Análisis de requerimientos de atributos de calidad.
- C) Diagrama de deployment (diseño arquitectónico funcional consistente con las respuestas anteriores) que satisfaga los requerimientos enunciados. Explicación de la estrategia para resolver el problema.

- D) Para satisfacer los requerimientos de atributos de calidad, el diseño arquitectónico debe refinarse con estilos y patrones de diseño adecuados para lograr la misión del negocio y satisfacer todos los requerimientos. Para cada patrón y estilo empleado, explicar debidamente el problema que resuelve.
- E) Informe en Word, con un diseño de experimentos ejecutado en el sistema, variando el número de consultas a ejecutar, el número de clientes, y el tiempo de ejecución total. También debe presentarse al análisis para encontrar el punto de corte para la distribución.