Sistema de Votación Universitaria Basado en Blockchain
INFO288 - Sistemas Distribuidos

Acádemico:
Dr. Luis Veas {luis.veasc@inf.uach.cl}
Instituto de Informática, Universidad Austral de Chile
Alumnos:
- Fernando Inzulza {fernando.inzulza@alumnos.uach.cl}
- Fernando Castillo {fernando.castillo@alumnos.uach.cl}
- Felipe Córdova {felipe.cordova@alumnos.uach.cl}
- Juan Santana {juan.santana@alumnos.uach.cl}
- Cristobal Pérez {cristobal.perez@alumnos.uach.cl}

31 de marzo de 2025

I. Introducción

I-A. Contexto

Los sistemas de votación tradicionales pueden presentar diversos desafíos, ya sea por falta de transparencia, seguridad o integridad de los votos. En muchos procesos de votación, estos podrían estar expuestos a fraudes o manipulación, provocando un sistema poco confiable para la toma de decisiones.

Además de estas vulnerabilidades, las votaciones presenciales presentan otros problemas, como la necesidad de formar largas filas [1], lo que genera pérdida de tiempo para los votantes. Esto provoca que muchas personas decidan no participar en el proceso electoral, afectando la representatividad de los resultados. A esto se suma el alto costo asociado a la impresión de papeletas y otros materiales físicos, impactando tanto en el presupuesto como en el medio ambiente.

En el ámbito universitario, las votaciones estudiantiles, ya sea para elecciones, paralizaciones o toma de decisiones, son fundamentales para la comunidad. Sin embargo, muchas instituciones utilizan herramientas digitales como Google Forms (Véase el Anexo A.) para llevarlas a cabo, lo que presenta inconvenientes importantes, como la falta de transparencia y la dificultad para verificar que los votantes sean realmente estudiantes.

Por estas razones, se propone el desarrollo de un sistema distribuido que garantice la seguridad e inmutabilidad de los votos, asegurando total transparencia. Para ello, se utilizará tecnología basada en ***blockchain***, la cual permite registrar y verificar los votos evitando posibles modificaciones o alteraciones en su registro. El voto electrónico puede ser definido como un sistema mediante el cual el votante registra directamente su o sus preferencias usando un dispositivo electrónico, ya sea una máquina diseñada específicamente para ello, una computadora personal o incluso un teléfono celular [2].

I-B. Objetivo

El objetivo principal de este proyecto es desarrollar un sistema de votación escalable y seguro utilizando tecnología blockchain, garantizando que cada usuario se verifique antes de emitir su voto y evitando cualquier posibilidad de que pueda votar más de una vez. Además, se busca facilitar la participación de la mayor cantidad de personas posible, eliminando las barreras de tiempo y logística que desincentivan la votación en procesos tradicionales.

Se espera que este sistema pueda ser adoptado no solo en entornos universitarios, sino también en pequeñas empresas que necesiten resguardar sus procesos de votación. A largo plazo, podría implementarse en votaciones municipales o incluso en elecciones gubernamentales, incluyendo elecciones presidenciales, asegurando procesos democráticos más confiables, seguros y eficientes.

II. Problema Detectado / Oportunidad de Innovación

Los sistemas de votación tradicionales presentan una serie de problemas que afectan la confiabilidad y eficiencia del proceso electoral:

1. **Falta de transparencia:** En muchos sistemas actuales, los votos no pueden ser verificados de manera independiente por los votantes, lo que puede generar desconfianza en los resultados. Esto se evidencio como una grave situación durante la jornada en una mesa de votación instalada en el local de Estación Mapocho. Una persona habría marcado votos, dificultando la votación y provocando el enojo de los votantes. [3]

2. **Manipulación:** Los sistemas centralizados pueden ser vulnerables a ataques como fraude electoral o alteración de resultados por actores malintencionados.
3. **Dificultad de la verificación:** La verificación de resultados puede requerir procesos largos y complejos, lo que retrasa la publicación de resultados.
4. **Accesibilidad:** La organización de elecciones tradicionales implica costos elevados y una logística compleja, especialmente en regiones remotas o con acceso limitado a infraestructura electoral.
5. **Seguridad de datos:** Los sistemas de votación electrónica convencionales pueden ser susceptibles a filtración de datos personales y suplantación de identidad.

Dada la importancia de la confiabilidad en los procesos democráticos, es fundamental encontrar una solución que aborde estos problemas de manera efectiva y escalable.

III. Solución Propuesta

Para abordar los problemas de los sistemas de votación convencionales, se propone el desarrollo de un sistema distribuido de votación basado en blockchain. Este enfoque permitirá mejorar la seguridad, transparencia y eficiencia del proceso electoral mediante las siguientes características clave:

1. **Inmutabilidad y Transparencia:** Gracias a la tecnología blockchain, cada voto quedará registrado de forma inalterable y accesible para su verificación por cualquier parte interesada, sin comprometer la privacidad del votante.
2. **Descentralización:** Al distribuir los registros en múltiples nodos, se elimina la dependencia de una única entidad central, reduciendo el riesgo de manipulación o fraude.
3. **Supervisión en Tiempo Real:** La naturaleza pública y verificable de blockchain permitirá a los organismos electorales y ciudadanos inspeccionar el conteo de votos en cualquier momento, agilizando la validación de los resultados.
4. **Seguridad Reforzada:** La utilización de contratos inteligentes garantizará que cada voto sea procesado de manera segura y sin intervención de terceros. Además, los mecanismos criptográficos protegerán la identidad y privacidad de los votantes.
5. **Accesibilidad y Eficiencia:** Un sistema basado en plataformas digitales reducirá los costos y la complejidad logística de las elecciones, facilitando la participación ciudadana incluso en regiones con infraestructura limitada.

Este enfoque busca modernizar el proceso electoral, ofreciendo un sistema confiable, seguro y accesible que refuerce la confianza pública en los resultados y optimice la gestión electoral.

IV. Diseño

IV-A. Diagrama de Arquitectura

El sistema propuesto utiliza una arquitectura basada en componentes que garantiza escalabilidad y seguridad. Esta arquitectura está organizada en cuatro capas principales:

1. **Capa del cliente - Frontend**: Aplicación donde los estudiantes podrán autentificarse para que puedan emitir su voto. Este se comunicara con el backend mediante APIs REST, permitiendo el envió de datos cifrados.
2. **Capa de la lógica - Backend**: Gestiona la lógica del sistema, con el uso de distintas tecnologías y módulos de autentificación, comunicación con base de datos, manejo de votos hacia la blockchain y balance de carga.
3. **Capa Blockchain**: Nodos descentralizados que almacenan los votos de forma inmutable (ej: red Ethereum de prueba).
4. **Capa de datos**: gestiona el almacenamiento, recuperación y control de datos

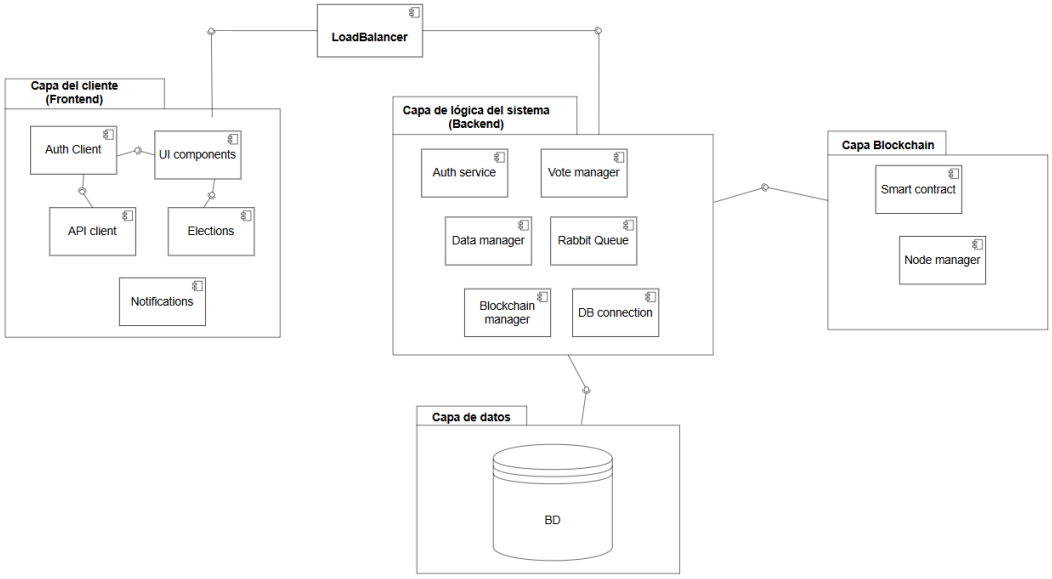

Figura 1: Diagrama de arquitectura basada en componentes

IV-A1. Flujo de Datos:

- **Autenticación**: El estudiante ingresa credenciales.
- **Emisión del voto**: El frontend envía el voto firmado al backend.
- **Validación inicial**: El backend verifica la firma y envía el voto a RabbitMQ.
- **Procesamiento asíncrono**:
  - Un worker consume el mensaje, interactúa con el contrato inteligente y registra el voto en la blockchain.
  - La transacción se guarda en un bloque y se emite un evento de confirmación.
- **Notificación**: El frontend recibe la confirmación vía WebSockets.

IV-B. Componentes de la Solución

Lista detallada de los componentes con sus descripciones:

- **Load Balancer**: Distribuye el tráfico entrante entre múltiples instancias para garantizar escalabilidad y disponibilidad.
- **Capa del Cliente (Frontend)**:
  - **Auth Client**: Módulo de autenticación y gestión de sesiones
  - **UI Components**: Componentes reutilizables para la interfaz de usuario
  - **API Client**: Cliente para comunicación con los servicios backend
  - **Elections**: Interfaz para visualización e interacción con procesos electorales
  - **Notifications**: Recibe la confirmación de votos y resultados
- **Capa de Lógica del Sistema (Backend)**:
  - **Auth Service**: Servicio de autenticación
  - **Vote Manager**: Coordina el proceso completo de votación
  - **Data Manager**: Gestiona el acceso a datos no blockchain
  - **Rabbit Queue**: Sistema de mensajería de colas
  - **Blockchain Manager**: Interactuar con la capa blockchain
  - **DB Connection**: Pool de conexiones a la base de datos tradicional
- **Capa Blockchain**:
  - **Smart Contract**: Implementa la lógica de votación en la blockchain (registro, verificación)
  - **Node Manager**: Administra la red descentralizada de nodos participantes
- **Capa de Datos**:
  - **BD**: Almacenamiento persistente para datos convencionales del sistema

IV-C. Modelo Físico (Infraestructura de Despliegue)

Los equipos donde se hará el despliegue del sistema sera en computadores personales de los estudiantes, algunos de estos tienen las siguientes características:

- **Equipo 1**: ASUS ROG STRIX G531GT, Intel Core i5-9300H de 9ª generación, con 4 núcleos y una velocidad base de 2.4 GHz, 16 GB DDR4 a 2666 MHz, Windows 11.
- **Equipo 2**: ASUS ROG STRIX B450-F GAMING II, AMD Ryzen 7 5700G de quinta generación, con 8 núcleos, velocidad de 3.8Ghz, 32GB DDR4 a 3200MHz hasta 3600MHz, Windows 10 Pro
- **Equipo 3**: HP Pavilion Laptop 15-cw1xxx, AMD Ryzen 5 3500U con Radeon Vega Mobile Graphics, con 4 núcleos y 8 hilos a una velocidad base de 2.1 GHz, 8 GB DDR4, Windows 11 Home.

IV-D. Modelo Fundamental

IV-D1. Arquitectura del Sistema: El sistema de votación propuesto se fundamenta en una arquitectura basada en componentes que permite:

- Separación de responsabilidades por componentes
- Escalabilidad
- Mantenibilidad del código
- Seguridad

Organizado en cuatro capas principales con interfaces bien definidas entre ellas.

IV-D2. Rendimiento del sistema: El sistema garantiza los siguientes parámetros de rendimiento:

- **Concurrencia**:
  - 500 usuarios simultáneos en proceso de votación activo
  - 1000 usuarios concurrentes en página principal
- **Capacidad**: Soporta hasta 10 elecciones concurrentes
- **Tiempos de respuesta**:
  - Máximo 10 minutos para ingreso de datos y confirmación de voto
  - 300 ms para almacenamiento en base de datos
  - 300 ms de carga entre página principal y elecciones

IV-D3. Requisitos No Funcionales:

- **Escalabilidad**: El sistema debe soportar picos de hasta 1000 usuarios concurrentes
- **Disponibilidad**: 99.9 % de disponibilidad durante períodos electorales
- **Seguridad**: Cifrado para todos los datos sensibles
- **Usabilidad**: Interfaz accesible para cualquier usuario

IV-E. Base de Datos

Descripción del modelo de datos propuesto para el sistema de votación:

IV-E1. Modelo Relacional: El diseño relacional consta de 5 entidades principales con las siguientes relaciones:

- **USUARIOS**: Almacena información de votantes y administradores.
- **ELECCIONES**: Registra los procesos electorales.
- **CANDIDATOS**: Contiene los participantes en cada elección.
- **VOTOS**: Traza cada voto emitido con integridad blockchain.
- **USUARIOS_PERMITIDOS**: Controla acceso a elecciones específicas.

Cuadro I: Esquema de tablas del modelo relacional

Tabla	**Atributos (PK en negrita, FK en cursiva)**
USUARIOS	**UUID**, rut, nombre, email, password, rol
ELECCIONES	**UUID**, título, descripción, start_date, end_date, hora_inicio, hora_fin, status, *UUID_user*
CANDIDATOS	**UUID**, nombre, descripción, *UUID_eleccion*, rut, email
VOTOS	**UUID**, date, blockchain_hash, *UUID_eleccion*, *UUID_user*
USUARIOS_PERMITIDOS	**UUID**, nombre, email, rut, *UUID_eleccion*

IV-E2. Relaciones: Las principales relaciones son:

- USUARIOS → ELECCIONES (1:N): Un usuario participa en múltiples elecciones.
- ELECCIONES → CANDIDATOS (1:N): Cada elección tiene varios candidatos.
- USUARIOS → VOTOS (1:N): Cada usuario puede emitir múltiples votos.
- ELECCIONES → USUARIOS_PERMITIDOS (1:N): Restricción de acceso por elección.

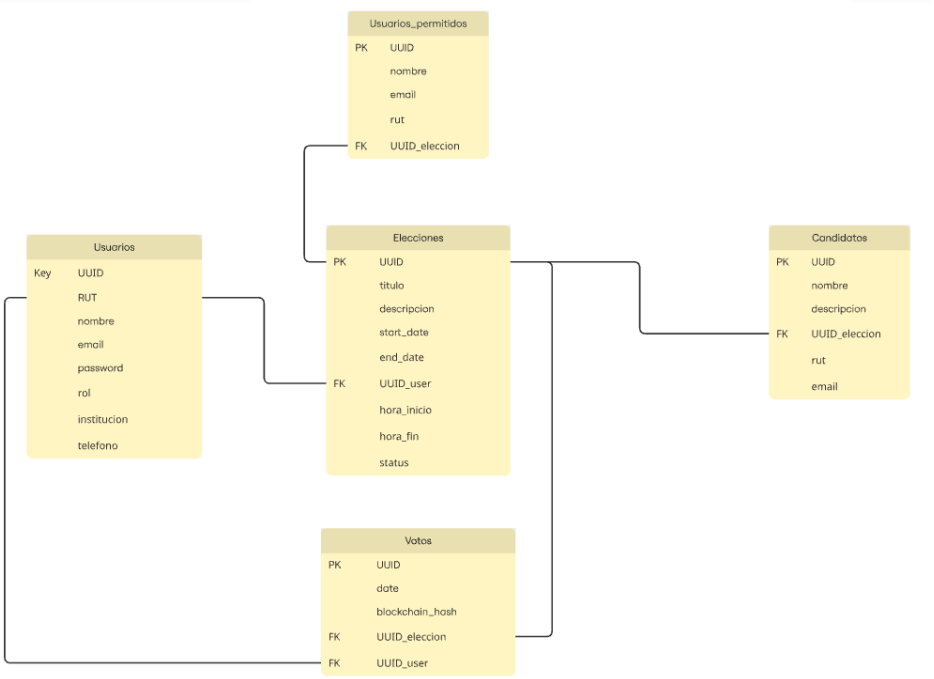

Figura 2: Diagrama Entidad-Relación del sistema

V. Consideraciones Técnicas

V-A. Lenguaje de Programación

Para la implementación del sistema de votación distribuido basado en blockchain, se han seleccionado los siguientes lenguajes y tecnologías:

V-A1. Frontend (Interfaz de Usuario):

- **Lenguaje:** JavaScript
- **Framework:** React 18.2 con Vite 5.3.10
- **Instalación:**

```
npm create vite@latest
```

- **Razón:** Se eligió React por su eficiencia en la construcción de aplicaciones web dinámicas, y Vite como herramienta de desarrollo por su rapidez en el arranque y recarga en caliente.

V-A2. Backend (API y Lógica de Negocio):

- **Lenguaje:** Node.js v22.14.0 (LTS) con Express
- **Instalación:**

```
npm install express
```

- **Razón:** Node.js permite manejar múltiples conexiones concurrentes, es eficiente para aplicaciones en tiempo real y facilita la integración con WebSockets y sistemas de colas como RabbitMQ.

V-A3. Contratos Inteligentes (Blockchain):

- **Lenguaje:** Solidity 0.8.23
- **Instalación:**

```
npm install --global solc
```

- **Razón:** Solidity es el lenguaje estándar para desarrollar contratos inteligentes en Ethereum y otras blockchains compatibles.

V-A4. Workers y Procesamiento Asíncrono:

- **Message Broker:** RabbitMQ 3.12

```
npm install amqplib
```

- **Razón:** Se encargada de procesar los votos encolados en RabbitMQ y registrarlos en la blockchain

V-A5. Base de Datos y Caché:

- **PostgreSQL:** 16.1
- **Instalación:**

```
sudo apt install postgresql
```

- **Razón:** Base de datos relacional para almacenamiento de información.

V-B. Herramientas de Software

- **Software Libre:**
  - React, Node.js, PostgreSQL, RabbitMQ
  - **Ventajas:** Cero costos de licencia
  - **Desventajas:** Requiere manejo de la tecnología para configuraciones avanzadas
- **Soluciones Blockchain Abiertas:**
  - Ethereum/Hyperledger
  - **Ventajas:** Descentralización, seguridad e inmutabilidad
  - **Desventajas:** Costos de transacciones

Decisión Final: Optamos por mayormente por Herramientas de Software libres con componentes propios para:

- Garantizar transparencia (requisito clave en sistemas electorales)
- Minimizar costos
- Mantener el control sobre la tecnología y lenguaje a usar

VI. Entrevista Técnica con el Cliente

Esta sección detalla las preguntas clave al cliente para definir los requisitos del sistema de votación blockchain

1. **¿Cuál es el volumen estimado de votantes concurrentes?**
   - Respuesta: 1,000 votantes simultáneos
2. **¿Qué pasa si el sistema se cae durante las elecciones?**
   - Respuesta: Se debe garantizar la disponibilidad en todo momento. En casos excepcionales, deberán existir mecanismo para verificar si el voto fue realmente efectuado y dar la posibilidad de repetir el proceso si no lo fue.
3. **¿Cómo evitan que una persona vote múltiples veces?**
   - Respuesta: Validamos tu identidad mediante RUT y bloqueamos intentos duplicados, tanto en la base de datos como en el smart contract de blockchain.
4. **¿Cuánto tiempo tengo para completar mi voto?**
   - Respuesta: Dispones de hasta 10 minutos desde que ingresas a la sección para efectuar el voto hasta confirmar.
5. **¿Puedo verificar que mi voto fue registrado correctamente?**
   - Respuesta: Sí, al finalizar recibirás un comprobante al email con un hash de transacción blockchain que puedes verificar.
6. **¿Qué requisitos técnicos necesito para votar?**
   - Solo necesitas un navegador web y Conexión a Internet.
7. **¿Cómo manejan elecciones múltiples?**
   - Respuesta: Soporta hasta 10 elecciones concurrentes, cada una con su propia configuración y datos.
8. **¿Cuál es su presupuesto para el desarrollo y mantenimiento del sistema?**
   - Respuesta: El presupuesto disponible es limitado por lo que el uso de tecnologías y servicios tiene que ser analizado muy detalladamente de tal forma de minimizar los costes.
9. **¿Cómo se asegura de que su sitio de votación sea escalable y capaz de manejar una gran cantidad de tráfico?**
   - Respuesta: Utilizando tecnologías como un orquestador de contenedores que permita replicar componentes o capas si se necesita.
10. **¿Cuál es el nivel de tráfico usual de su sitio web? ¿Ha notado algún aumento de la demanda en algún momento en particular?**
    - Respuesta: El tráfico usual del sitio es de menos de 100 personas en un día normal, este se ve aumentado en momentos particulares como toma de decisiones en alguna asamblea.

Cuadro II: Desglose de costos mensuales en AWS para sistema de votación estudiantil

Componente	**Servicio AWS**	**Costo (USD/mes)**
Frontend (React)	S3 + CloudFront	$20
Backend (Node.js)	Lambda + API Gateway	$10
Base de Datos	RDS PostgreSQL	$20
Blockchain	EC2 (Hyperledger)	$70
Workers	SQS + Lambda	$5
Total		**$125**

VII. Análisis de Costos y Optimización de Recursos

Considerando que la organización no cuenta con infinitos recursos, por ende, espera aprovechar sus recursos de la mejor manera posible, se presentan la alternativa de implementación con sus respectivos análisis de costos.

VIII. Anexos

VIII-A. Anexo A

- **Correo Centro Estudiantes Informática:** A continuación se presenta una captura de pantalla del correo UACh, mostrando una votación del año 2023 mediante Google Forms.

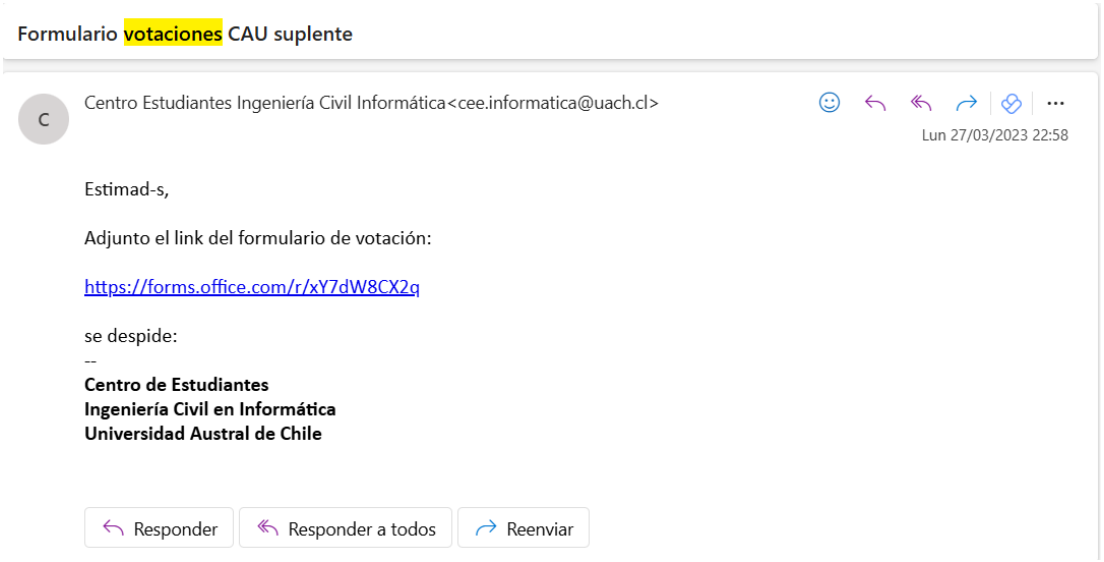

Figura 3: Votación online por Google Form

REFERENCIAS

[1] T13. Elecciones 2024: Gobierno explica las posibles razones de las largas filas en locales. Disponible en: https://www.t13.cl/noticia/elecciones-2024/elecciones-2024-gobierno-explica-las-posibles-razones-las-largas-filas-locales-26-10-2024.

[2] J. Smith. International experiences of electronic voting and their implications for new south wales, 2009. Recuperado de https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0719-27892018000200012#B49.

[3] ADN Radio. Reportan posible fraude electoral en estación mapocho. Disponible en: https://www.adnradio.cl/2024/10/26/reportan-grave-situacion-acusan-que-apoderado-de-mesa-marco-votos-en-una-mesa-de-estacion-mapocho/.

[4] CHV Noticias. Transmisión en vivo: Jornada de elecciones 2024. Disponible en: https://www.youtube.com/watch?v=w7fE7nDMdsg.

[5] T13. Elecciones 2024: Gobierno explica las posibles razones de las largas filas en locales. Disponible en: https://www.t13.cl/noticia/elecciones-2024/elecciones-2024-gobierno-explica-las-posibles-razones-las-largas-filas-locales-26-10-2024.

[6] T13. Tenso cierre de mesas en huechuraba: Denuncian cajas mal selladas y manipuladas por personas no autorizadas. Disponible en: https://www.t13.cl/noticia/nacional/tenso-cierre-mesas-huechuraba-denuncian-cajas-mal-selladas-manipuladas-por-pers-26-10-2024.