



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

TALLER DE PROYECTO 2 INGENIERÍA EN COMPUTACIÓN

G3 – SmartPoll: Sistema de votación segura y transparente con blockchain

Informe Final

Grupo de trabajo:

- Blasco, Gonzalo Gabino - Legajo N° 03282/6
- Cabral, Ramiro Nicolás - Legajo N° 03226/6
- Polanis, Iván Valentín - Legajo N° 03266/5

Docente responsable: Gastón Maron



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

Índice General

1. Descripción General del Proyecto.....	4
1.a. Propuesta original.....	4
1.b. Propuestas o modificaciones a partir de evaluaciones de la cátedra.....	4
1.c. Decisiones propias o complementarias.....	5
2. Presentación Esquemática del Proyecto.....	6
2.1. Esquema de procesos.....	6
2.2. Esquema de conexiones.....	8
2.3. Funcionalidades y requerimientos.....	10
3. Documentación del Software del Proyecto.....	12
3.1 Frontend.....	14
3.2 Backend.....	15
3.3 Infraestructura en AWS (infra).....	18
4. Otra Documentación Relacionada.....	21
4.1. Hardware construido.....	21
4.2. Interfaz de Escaneo de QR.....	22
4.3. Interfaz de votación.....	25
4.4. Interfaz de auditoría.....	26
4.5. Interfaz de visualización de la Blockchain.....	28
Apéndice A: Materiales y Presupuesto.....	30



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

Índice de Figuras

Figura 1: Flujo de funcionamiento de SmartPoll.....	6
Figura 2: Esquema de conexiones de hardware en mesa de entrada.....	9
Figura 3: Esquema de conexiones de hardware en cuarto oscuro.....	10
Figura 4: Esquema de comunicación entre las aplicaciones.....	14
Figura 5: Arquitectura en la nube.....	18
Figura 6: Diagrama de componentes de la red.....	20
Figura 7: Materiales utilizados para la lectura del código QR.....	21
Figura 8: Interfaz Web del inicio de sesión de SmartPoll.....	22
Figura 9: Interfaz Web QR-Pase.....	23
Figura 10: Interfaz Web QR-Pase escaneado con éxito.....	24
Figura 11: Interfaz de votación de SmartPoll en estado de espera.....	25
Figura 12: Interfaz de votación de SmartPoll con los candidatos.....	26
Figura 13: Interfaz de auditoría de SmartPoll.....	27
Figura 14: Interfaz de visualización de la blockchain.....	28

Índice de Tablas

Tabla 1: Conexionado de Raspberry Pi y módulo I2C.....	10
Tabla A.1: Materiales utilizados en SmartPoll.....	30



FACULTAD DE INFORMÁTICA

1. Descripción General del Proyecto

El proyecto **SmartPoll** propone el desarrollo de un sistema de votación electrónica basado en blockchain, con el objetivo de superar las limitaciones de los métodos tradicionales y de otras tecnologías previas de voto electrónico. El enfoque principal es garantizar transparencia, verificabilidad y confianza en el proceso electoral, minimizando riesgos de fraude y preservando la privacidad del votante.

La propuesta se centra en implementar un flujo de votación controlado mediante pases de acceso digitales. Cada votante recibe un código QR único, aleatorio y firmado, que valida su derecho a participar. Una vez autenticado, el sistema emite un token anónimo de voto, independiente del QR inicial, que se consume al emitir el sufragio, evitando el doble voto y asegurando la separación entre identidad e intención de voto. El registro final se realiza en una blockchain permissionada, garantizando inmutabilidad, auditoría abierta y conteo verificable.

1.a. Propuesta original

El diseño inicial incluyó:

- Generación de códigos QR únicos asociados al DNI para autenticar votantes.
- Validación de dichos QR mediante un sistema de ingreso seguro.
- Emisión de un token de voto anónimo tras la validación.
- Registro de cada sufragio como transacción en una blockchain permissionada.
- Conteo automático, público y verificable de los resultados.
- Separación estricta entre la identidad del votante y el contenido del voto.

Se definieron como **objetivos primarios** el desarrollo de la interfaz web, el mecanismo de validación, la interfaz de sufragio y la transparencia del proceso; y como **objetivos secundarios** la posibilidad de soportar múltiples votaciones y el padrón electoral.

1.b. Propuestas o modificaciones a partir de evaluaciones de la cátedra

Hasta el momento, la **propuesta original se mantiene intacta**, sin cambios solicitados por la cátedra en esta primera etapa.



FACULTAD DE INFORMÁTICA

1.c. Decisiones propias o complementarias

Durante el desarrollo se modificó el flujo original con el fin de reforzar la seguridad y simplificar la experiencia del votante:

- Se resolvió que el votante solo pueda presentarse con un dispositivo móvil con conexión a internet. Esta decisión elimina el uso de QR impresos, reduciendo riesgos de pérdida, duplicación o falsificación, y asegura que toda la comunicación del pase se realice en forma digital.
- Se decidió que el Token Anónimo de Voto (TAV) no sea entregado al votante, sino generado directamente en la estación de votación. De esta manera, se evita que el votante transporte el TAV y se garantiza que solo se utilice dentro del entorno controlado de la mesa. En este nuevo esquema, es la estación de votación la que habilita al dispositivo de la mesa de ingreso para que se pueda escanear un QR. Una vez escaneado y validado el pase, el dispositivo de ingreso notifica a la estación de votación que se autorice una nueva votación y, en ese momento, se genera el TAV en la propia estación.
- Se decidió realizar el deploy completo de la plataforma en la nube, utilizando Amazon Web Services (AWS) como entorno de producción. Esta decisión tuvo como finalidad asegurar la disponibilidad, estabilidad y coherencia entre los distintos módulos del sistema, permitiendo que tanto el backend como el frontend funcionen de manera integrada bajo condiciones reales de operación.
- Se incorporó una pantalla LCD en la mesa de entrada, conectada a la Raspberry Pi, que permite informar al votante el resultado del escaneo de su código QR. De este modo, el votante no depende de tener la aplicación abierta para conocer el estado del proceso y, además, es posible realizar la verificación utilizando un QR impreso en papel.
- Se añadió un temporizador destinado a evitar que una persona abandone el cuarto oscuro sin completar la selección de un candidato, lo que podría bloquear el sistema. El tiempo máximo de este temporizador es configurable y puede ajustarse según las necesidades de la implementación.

FACULTAD DE INFORMÁTICA

2. Presentación Esquemática del Proyecto

2.1. Esquema de procesos

En la **Figura 1** se representa gráficamente el flujo de funcionamiento de SmartPoll.

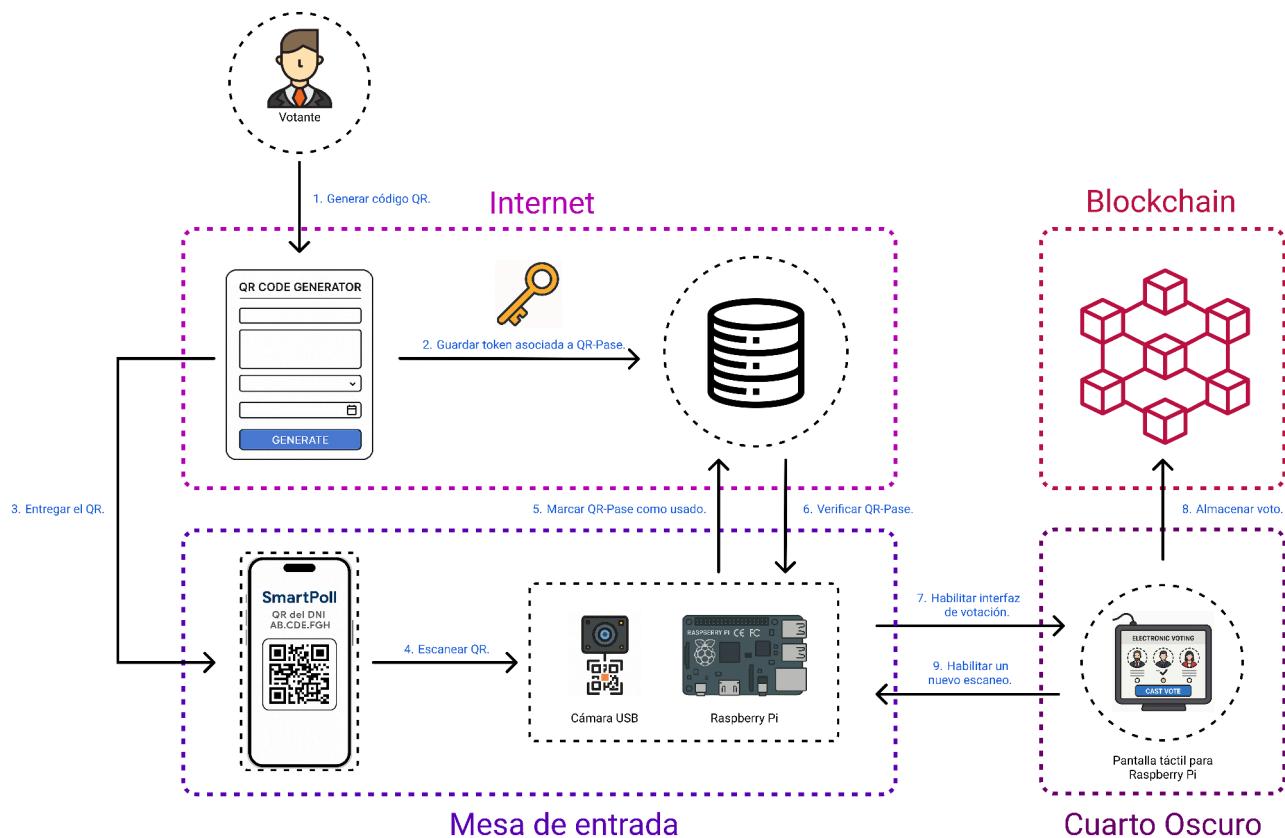


Figura 1: Flujo de funcionamiento de SmartPoll.

A continuación, se detalla cada una de las etapas del flujo representado en la **Figura 1**, describiendo las acciones que realiza el votante y el hardware involucrado en cada paso.

Antes de iniciar el proceso, el votante debe registrarse en la plataforma web, ingresando correo electrónico, DNI y contraseña.

1. Generación del QR-Pase (Plataforma Web + Servidor en la nube):

Una vez validado el registro, el sistema genera un token aleatorio y lo asocia a



FACULTAD DE INFORMÁTICA

un QR-Pase único y firmado, vinculado al DNI del votante. El QR se muestra en la plataforma para que el votante pueda presentarlo en la mesa de ingreso.

2. **Almacenamiento del token asociado al QR-Pase (Base de datos en la nube):**
El token generado se guarda en la base de datos en la nube, quedando listo para ser validado en el momento del sufragio.
3. **Presentación del QR-Pase (Dispositivo del votante):**
El votante presenta el QR-Pase en su dispositivo en la mesa de entrada para su verificación.
4. **Escaneo del QR-Pase (Raspberry Pi con cámara USB):**
Una Raspberry Pi equipada con cámara USB escanea el QR-Pase presentado por el votante.
5. **Marcado del QR-Pase como utilizado (Raspberry Pi + Servidor en la nube):**
La Raspberry Pi valida contra la base de datos si el QR-Pase es válido y no ha sido consumido.
 - Si es válido, el sistema lo marca como utilizado.
 - Si no es válido o ya fue consumido, no se permite el acceso al cuarto oscuro.
6. **Respuesta del servidor (Servidor + SSE activo):**
A través de la conexión SSE, el servidor notifica al cliente que el QR-Pase ya fue consumido. El dispositivo del votante muestra que está habilitado para ingresar al cuarto oscuro y la conexión se cierra.
7. **Habilitar Interfaz de votación: (Raspberry Pi en cuarto oscuro):**
Cuando la Raspberry de la mesa de entrada notifica el ingreso de un votante, la Raspberry del cuarto oscuro genera y firma el Token Anónimo de Votación (TAV), habilitando la interfaz de votación. El votante emite su voto, que se envía junto con el TAV firmado al servidor para su validación y registro.
8. **Almacenamiento del TAV en la blockchain permisionada (Servidor + Red blockchain):**
El TAV se registra en la blockchain permisionada, asegurando inmutabilidad, trazabilidad y conteo verificable.



FACULTAD DE INFORMÁTICA

9. Habilitar un nuevo escaneo (Raspberry Pi en cuarto oscuro):

Una vez finalizada una votación, la Raspberry Pi del cuarto oscuro se comunica con la de la mesa de entrada para permitir un nuevo escaneo de QR.

2.2. Esquema de conexiones

El mismo se caracteriza por su sencillez y modularidad, dado que solo intervienen dos unidades de Raspberry Pi 3 Model B, cada una con un propósito bien definido:

- **Raspberry Pi de Mesa de Ingreso:** conectada a una cámara USB Logitech C170, encargada de la lectura del QR-Pase presentado por el votante. Además, tiene conectada una pantalla LCD para notificar al usuario cuando el QR fue escaneado correctamente o si ocurrió algún error.
- **Raspberry Pi del Cuarto Oscuro:** conectada a una pantalla táctil, utilizada para la interfaz de votación y emisión del sufragio.

Ambas placas se alimentan mediante un adaptador de 5V y 3A, y se comunican con el servidor alojado en la nube mediante conexión Ethernet o Wi-Fi, dependiendo de la disponibilidad de red.

El esquema de conexiones físicas del sistema SmartPoll se puede dividir en las conexiones para la mesa de entrada y el cuarto oscuro.

En la **Figura 2** puede observarse el esquema de conexionado de la mesa de entrada:

FACULTAD DE INFORMÁTICA

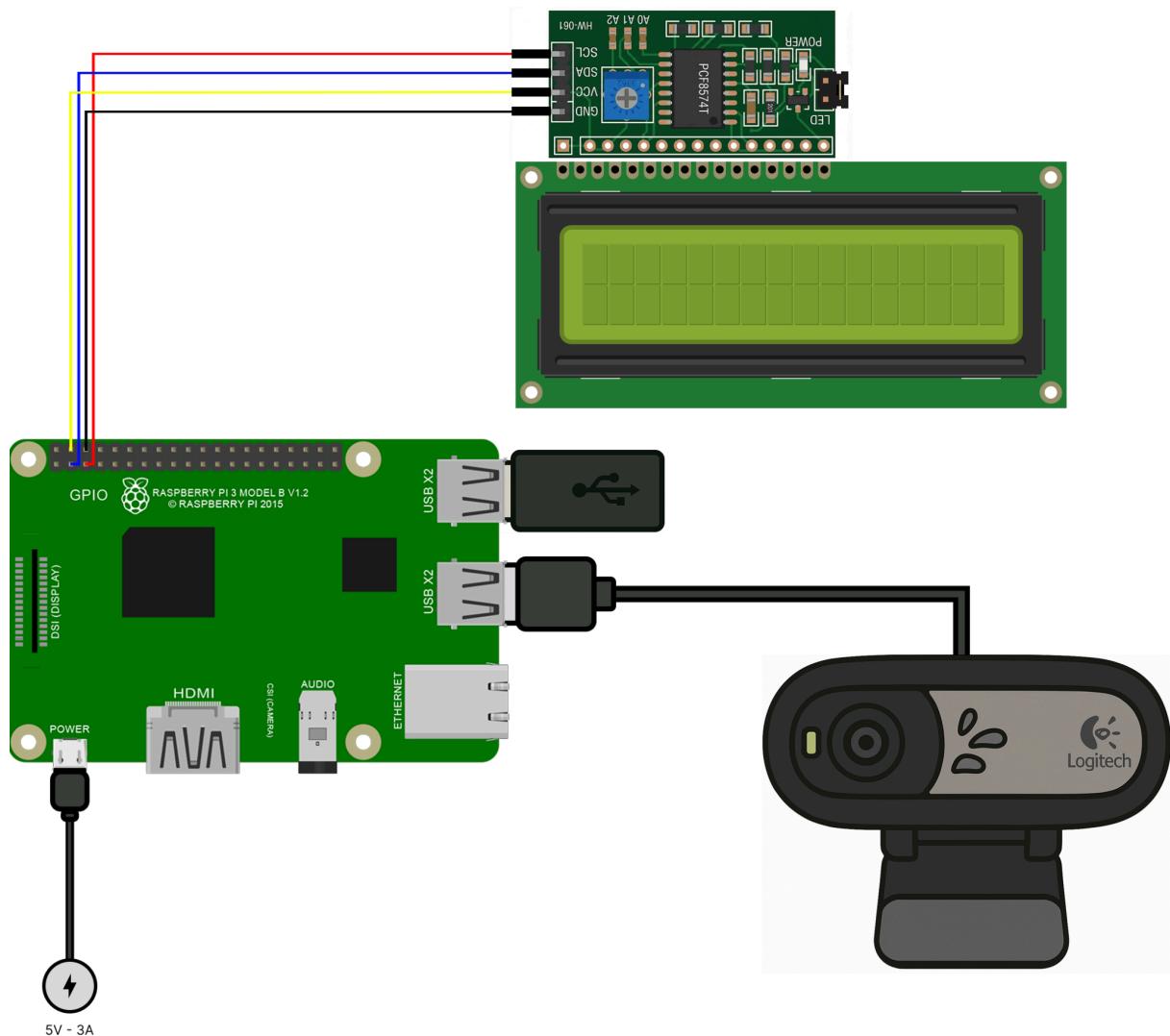


Figura 2: Esquema de conexiones de hardware en mesa de entrada.

En la misma puede observarse la alimentación anteriormente detallada, así como la conexión con la cámara mediante un cable USB. Además se tiene conectado un módulo I2C y un LCD para informar al usuario. El conexionado a la Raspberry Pi se puede observar en la **Tabla 1**.

FACULTAD DE INFORMÁTICA

Pin módulo I2C	Pin Raspberry Pi
GND	Pin 6
VCC	Pin 4
SDA	Pin 3
SCL	Pin 5

Tabla 1: Conexionado de Raspberry Pi y módulo I2C.

En la **Figura 3** se muestra el esquema de conexión para la Raspberry Pi del cuarto oscuro:

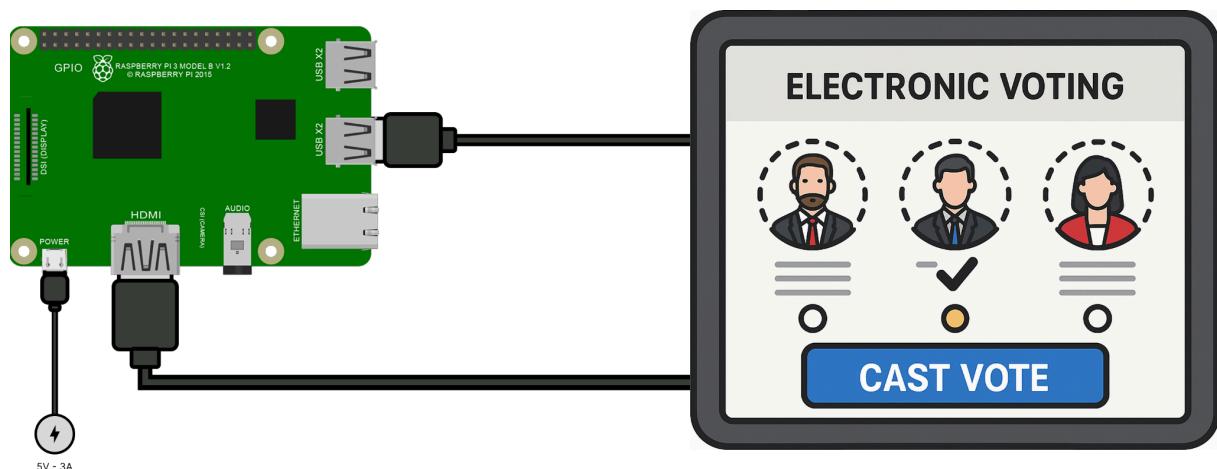


Figura 3: Esquema de conexiones de hardware en cuarto oscuro.



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

Se destaca la conexión de la pantalla táctil al sistema mediante dos interfaces principales: un cable HDMI, encargado de la transmisión de la señal de video desde la Raspberry Pi hacia la pantalla, y un cable USB, que establece el canal de comunicación necesario para detectar y procesar las acciones táctiles del usuario.

2.3. Funcionalidades y requerimientos

2.3.1. Requerimientos funcionales

Los requerimientos funcionales mencionados en las entregas anteriores se listan a continuación:

1. Permitir a los usuarios generar un código QR único asociado a su DNI para autenticación.
2. Validar el QR presentado por un votante antes de habilitarlo para emitir el sufragio.
3. Ofrecer una interfaz de votación sencilla donde el usuario pueda seleccionar una opción y enviarla.
4. Emitir un token anónimo de voto al escanear el QR una vez validado el mismo.
5. Registrar cada voto en la blockchain de manera anónima, asegurando su inmutabilidad.
6. Asegurar que cada votante pueda votar una sola vez en cada elección.
7. Realizar el conteo de votos automáticamente al finalizar el período de votación y mostrarlo de forma pública y verificable.
8. Posibilitar que un usuario genere un código QR para participar en múltiples votaciones distintas.
9. Implementar un mecanismo de validación de elegibilidad, que determine qué votantes están habilitados en cada elección.

Los mismos se cumplieron, excepto por los requerimientos 8 y 9, que de igual forma corresponden a objetivos secundarios.



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

2.3.2. Requerimientos no funcionales

Los requerimientos no funcionales detallados en los informes previos se enumeran a continuación:

1. Diseñar interfaces responsivas, que puedan visualizarse desde distintos dispositivos.
2. Proteger contra intentos de voto múltiple, manipulación de datos o acceso no autorizado.
3. Separar estrictamente entre identidad del votante y contenido del voto.
4. Asegurar la disponibilidad de los registros de votación en blockchain para verificación pública.
5. Confirmación inmediata de autenticación, emisión y registro de cada voto.
6. Operar de forma consistente y sin pérdidas de información a lo largo de todo el proceso electoral.
7. Uso de Raspberry Pi como plataforma de hardware para la mesa de entrada y los dispositivos de votación en el cuarto oscuro.
8. La lectura del código QR debe realizarse con una cámara conectada a una Raspberry Pi.
9. Permitir la auditoría del proceso mediante acceso abierto al código fuente (open source) y a los registros en la blockchain.
10. La blockchain debe ser permisionada.

Todos los requerimientos fueron satisfactoriamente cumplidos. Sin embargo, por cuestiones operativas, los puntos 4 y 9 se encuentran parcialmente completos, dado que la blockchain está alojada en una red local. En un entorno de despliegue más realista, ambos requerimientos podrían cumplirse plenamente.



FACULTAD DE INFORMÁTICA

3. Documentación del Software del Proyecto

A continuación se detalla el propósito y el contenido de cada carpeta presente en el repositorio del código fuente del proyecto:

- **.github/workflows:** Contiene los archivos de configuración de GitHub Actions utilizadas en el proceso de CI/CD del sistema de generación de códigos QR. Incluye dos flujos de trabajo distintos para el despliegue en AWS: uno para el frontend (desplegado en Amazon S3) y otro para el backend (desplegado en Amazon ECS).
- **audit-app:** Contiene los archivos de la aplicación web destinada a la auditoría de los votos almacenados en la blockchain.
- **backend:** Incluye la API responsable de la creación y almacenamiento de los códigos QR.
- **blockchain:** Contiene la configuración y los archivos necesarios para el despliegue de la red Hyperledger Fabric, utilizada para el almacenamiento de los votos.
- **infra:** Alberga los archivos de Infraestructura como Código (IaC) empleados para la creación y configuración de la infraestructura en AWS.
- **qr-access-app:** Contiene los archivos del *frontend* de la web de generación de códigos QR.
- **qr-scan:** Incluye el script en Python ejecutado en la Raspberry Pi de la mesa de entrada, encargado de escanear los códigos QR generados por los usuarios.
- **votation-kiosk:** Contiene los archivos del frontend de la interfaz de votación.
- **votation-server:** Incluye el *Backend* correspondiente a la interfaz de votación. Es el responsable de la emisión de votos a la blockchain y quien audita a la misma.

La comunicación de estos módulos puede observarse en la **Figura 4**:

FACULTAD DE INFORMÁTICA

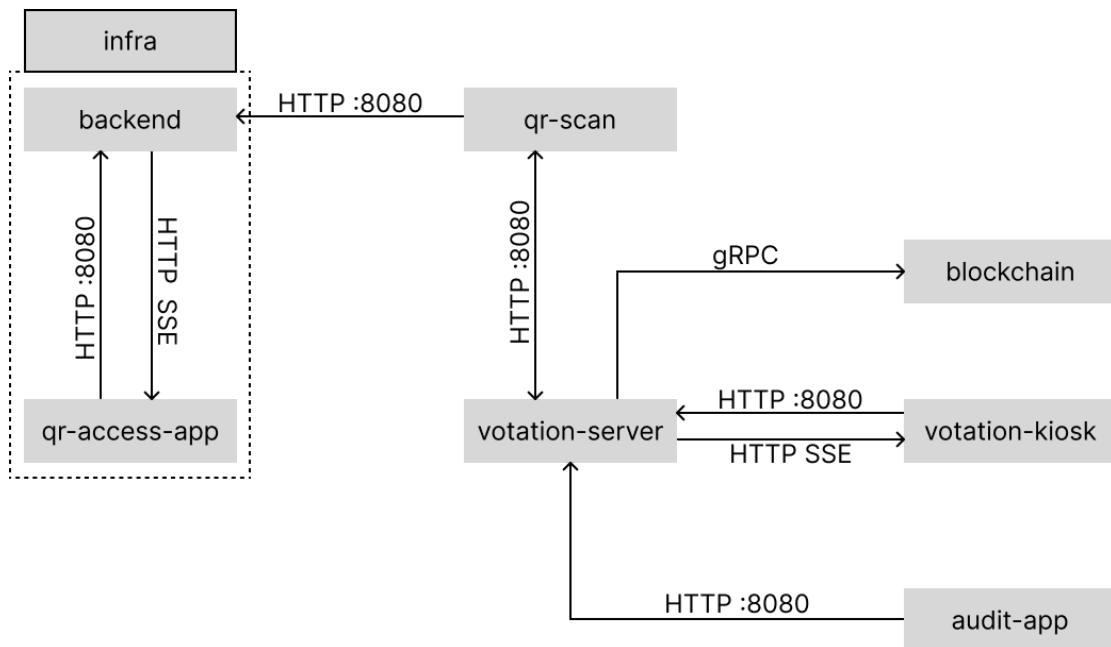


Figura 4: Esquema de comunicación entre las aplicaciones.

El backend, alojado dentro del entorno infra, gestiona la generación y validación de los códigos QR y se comunica con el frontend qr-access-app mediante HTTP y SSE.

El módulo qr-scan, ejecutado en la Raspberry Pi de la mesa de ingreso, valida los QR contra el backend y notifica al votation-server, encargado de coordinar el proceso de votación.

Este servidor se comunica con la interfaz de votación en el kiosk mediante Server-Sent Events (SSE), permitiendo notificaciones en tiempo real y una comunicación unidireccional desde el servidor hacia el cliente.

3.1 Frontend

Para todas las aplicaciones de interfaz (interfaz de votación, generador de QR y panel de auditoría) se mantuvo un stack tecnológico unificado, lo que permitió asegurar la coherencia visual, funcional y de desarrollo en todo el proyecto.

Se utilizó React junto con Vite y TypeScript, combinados con SWC como compilador. Esta configuración ofrece tiempos de compilación reducidos y tipado estático.

El entorno estándar fue Node.js 22.19 LTS, ejecutado dentro de contenedores Docker, garantizando entornos reproducibles y consistentes entre los distintos desarrolladores.



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

Para el maquetado y diseño visual se emplearon TailwindCSS, Radix-UI y ShadCn, lo que permitió construir componentes estilizados, accesibles y coherentes entre todas las aplicaciones, manteniendo un estilo uniforme y evitando duplicación de código.

La gestión de dependencias se realizó mediante pnpm, priorizando velocidad y eficiencia en el uso de espacio. La comunicación con los backends se implementó mediante HTTP/REST, mientras que la sincronización en tiempo real se resolvió con Server-Sent Events (SSE), elegido por su simplicidad en comunicaciones unidireccionales y capacidad de autoreconexión ante fallos de red.

3.1.1 Interfaz de auditoría (audit-app)

Aplicación destinada a la visualización de resultados de votación y auditorías. Se comunica con el votation-server mediante HTTP/REST para consultar el estado de la votación y los resultados en tiempo real.

Se implementaron componentes reutilizables para mostrar tablas de resultados, gráficos y alertas de auditoría, permitiendo que la misma arquitectura pueda adaptarse a futuras extensiones del sistema.

3.1.2 Interfaz de escaneo de QR-Pase (qr-access-app)

Aplicación utilizada por el votante para validar identidad y presentar el QR-Pase en la mesa de entrada. Permite indicar visualmente al usuario cuando el QR ha sido validado y está habilitado para emitir el voto.

La aplicación mantiene una comunicación constante con el backend a través de SSE, asegurando que los datos del votante se mantengan consistentes mientras navega por la interfaz.

3.1.3 Interfaz de votación (votation-kiosk)

Aplicación ejecutada en la Raspberry Pi del cuarto oscuro, cuyo objetivo principal es permitir la emisión del voto de manera segura y sincronizada.

Se comunica con el votation-server mediante SSE, lo que permite que el servidor envíe actualizaciones en tiempo real al kiosk, notificando cuando la votación está habilitada o finalizada.

La elección de SSE sobre websockets se fundamenta en que la comunicación es principalmente unidireccional, reduciendo complejidad y consumo de recursos,



FACULTAD DE INFORMÁTICA

mientras que la autoreconexión nativa de SSE garantiza robustez ante interrupciones de la red.

El kiosk mantiene un estado local del votante, gestionado mediante hooks de React, asegurando que cada votante pueda emitir su sufragio de manera independiente y que el sistema sea tolerante a errores o reinicios temporales.

3.2 Backend

3.2.1 Backend de gestión de usuarios (backend)

El backend de gestión de usuarios se desarrolló utilizando Spring Boot 3 sobre Java 21 LTS y se encarga de la gestión de usuarios, generación de códigos QR y almacenamiento de información asociada. Su responsabilidad es garantizar la integridad de las credenciales de los votantes y la correcta emisión de los QR-Pase utilizados para habilitar la votación.

El servicio expone endpoints HTTP que permiten a la Raspberry Pi de la mesa de entrada (qr-scan) validar los códigos QR mediante la cámara conectada. Una vez que un QR es validado, la aplicación de votación puede habilitar al votante para emitir su sufragio. Este backend fue contenerizado con Docker y desplegado en AWS ECS, permitiendo que el entorno de producción sea consistente y reproducible.

3.2.2 Backend de votación (votation-server)

El webserver se desarrolló con Node.js y Express y actúa como el núcleo de la gestión de Tokens Anónimos de Votación (TAVs) y la conexión con la blockchain. Este servicio se encarga de generar los TAVs, almacenarlos temporalmente en Redis con TTL para garantizar que expiren si no se usan, y de registrar los votos emitidos en la blockchain permisionada.

La comunicación con la interfaz de votación en el kiosk se realiza mediante Server-Sent Events (SSE), permitiendo notificaciones en tiempo real que habilitan la votación. Una vez que el votante emite su sufragio, el webserver lo envía a la blockchain mediante gRPC, utilizando las credenciales locales de la organización administradora que corren en la misma máquina. Este sistema de autenticación garantiza que solo usuarios autorizados puedan registrar votos en la red blockchain.



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

El diseño permite escalar la operación a otras organizaciones si fuese necesario, compartiendo las credenciales adecuadas para operar como la entidad administradora dentro de la red, manteniendo la integridad y seguridad de las transacciones.

El webserver mantiene la lógica de auditoría y validación de votos, permitiendo que los distintos componentes del sistema interactúen de manera coordinada y segura. El servicio se ejecuta en Docker en la Raspberry Pi del cuarto oscuro.

3.2.3 Backend de escaneo de QR-Pase (qr-scan)

El tercer servicio, desarrollado en Python con Flask, se encarga de detectar y decodificar los códigos QR capturados por la cámara de la Raspberry Pi, interpretando la información contenida en ellos para iniciar o validar una votación. Además, se encarga de informar al webserver sobre cada acción dentro del proceso de votación.

No realiza lógica compleja ni almacenamiento persistente, sino que actúa como un puente que comunica los eventos del kiosk y del backend de gestión de usuarios con el Webserver. De esta forma, se asegura que el flujo de votación se mantenga coordinado, permitiendo la participación de múltiples votantes de manera segura.

FACULTAD DE INFORMÁTICA

3.3 Infraestructura en AWS (infra)

La totalidad del entorno cloud se gestionó con Terraform, siguiendo un diseño modular que nos permite mantener la infraestructura organizada y escalable. Cada módulo representa un componente específico del entorno, con el fin de permitir su reutilización y la aplicación de cambios controlados.

Se implementaron módulos independientes para los principales servicios: ECR (repositorios de imágenes Docker), ECS (orquestación de contenedores), S3 (almacenamiento de archivos estáticos), ELB (balanceo de carga) y VPC (red privada virtual). Luego, se incorporaron RDS para la base de datos y CloudFront para la distribución de contenido. Esta arquitectura puede observarse en la **Figura 5**:

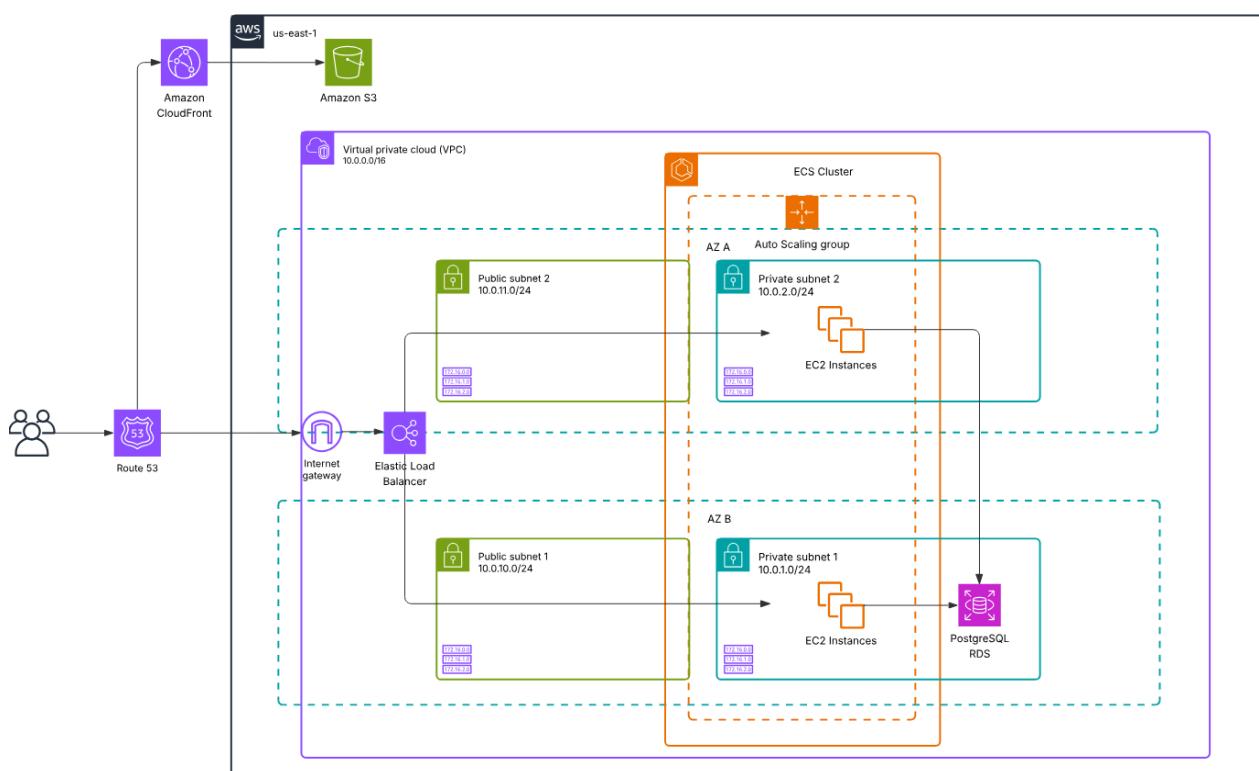


Figura 5: Arquitectura en la nube.

Dado que la infraestructura del proyecto posee una complejidad limitada, el archivo de estado (state file) de Terraform se almacenó localmente, evitando la necesidad de un backend remoto. Esta decisión simplificó el proceso de desarrollo y despliegue, manteniendo una configuración ligera y adecuada para el alcance del actual proyecto.



FACULTAD DE INFORMÁTICA

3.4 Blockchain (blockchain)

La integración del sistema SmartPoll con Hyperledger Fabric permitió garantizar la emisión, almacenamiento y auditoría de votos sobre una blockchain permisionada, asegurando trazabilidad, transparencia e integridad del proceso electoral.

La red de Fabric y el webserver basado en Express se ejecutan en contenedores Docker separados, comunicándose a través de la red interna. Esta configuración permite al webserver acceder directamente a las credenciales y certificados de la organización, evitando la dependencia de un sistema de identidad externo y replicando el comportamiento de un despliegue distribuido real.

Se desarrolló el contrato inteligente VoteContract, implementado en TypeScript, que define las operaciones principales del sistema: creación de votos, lectura del ledger, verificación de existencia de votos y conteo total. El contrato implementa un control de permisos, restringiendo la creación y conteo de votos exclusivamente a la organización administradora, mientras que el resto de nodos participantes pueden mantener solo capacidad de lectura y verificación. Esto asegura la integridad de los resultados y evita manipulaciones indebidas en el ledger.

Se adoptaron diversas decisiones de diseño para garantizar la robustez y escalabilidad del sistema. La separación de contenedores entre el webserver y la red de blockchain permite reproducir un esquema de despliegue real, donde la infraestructura y el servidor de aplicación pueden residir en dominios o máquinas diferentes, manteniendo independencia y modularidad. Además, cada elección se ejecuta en su propio canal, lo que aísla los procesos electorales y facilita auditorías posteriores. Aunque actualmente todo se ejecuta en la misma máquina, la comunicación entre webserver y blockchain se realiza mediante certificados y gRPC, replicando fielmente el comportamiento de un entorno distribuido y asegurando seguridad y trazabilidad en todas las transacciones.

Esta arquitectura asegura que cada voto sea registrado de manera inmutable y verificable, y que el sistema pueda escalar a múltiples organizaciones y elecciones manteniendo seguridad y transparencia en toda la plataforma.

En la **Figura 6** se muestra cómo está organizada la estructura de la blockchain utilizada, detallando la conexión entre sus distintos componentes.

FACULTAD DE INFORMÁTICA

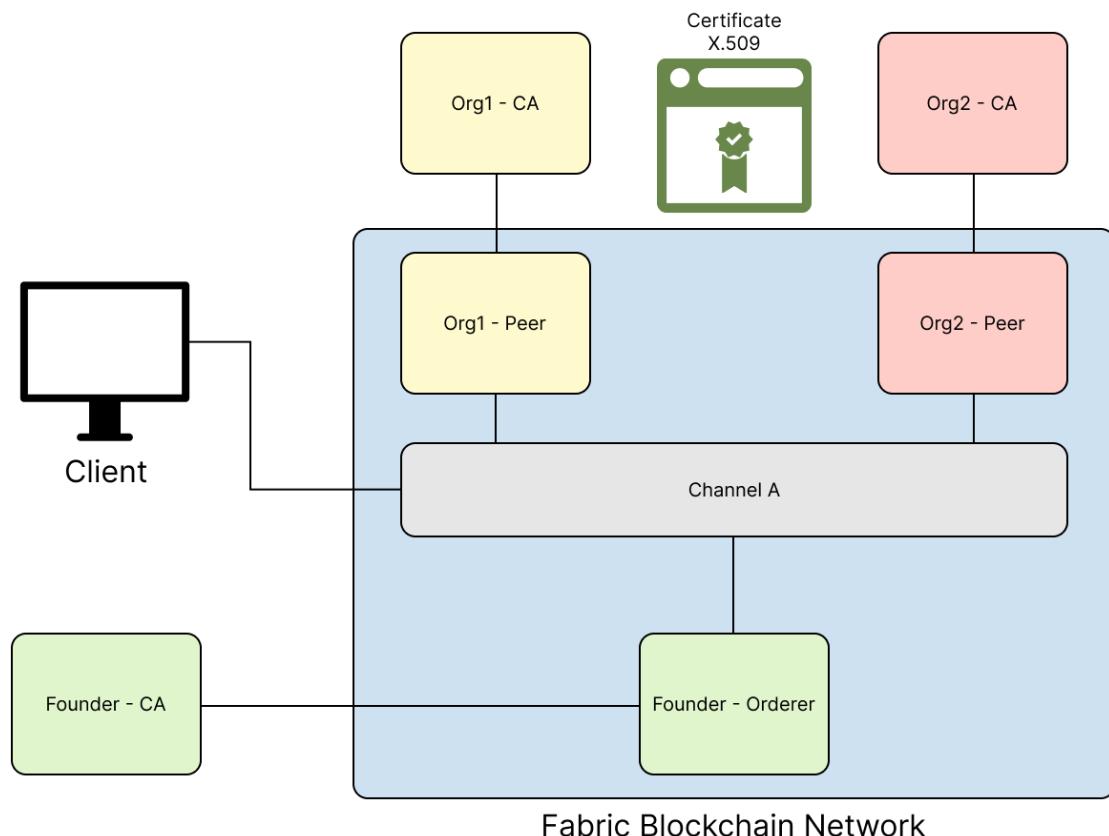


Figura 6: Diagrama de componentes de la red.

La figura muestra un ejemplo de cómo se constituye la red, compuesta por tres organizaciones: Founder, Org1 y Org2, cada una con su Autoridad Certificadora (CA) y su respectivo Peer, encargados de validar y mantener el estado del ledger. El Founder además ejecuta el nodo Orderer, responsable de ordenar las transacciones y garantizar la coherencia del canal de comunicación Channel A, donde se registran los votos.

El cliente, que representa la aplicación SmartPoll, interactúa con la red a través de certificados X.509, asegurando autenticidad y trazabilidad en cada operación. Cabe destacar que esta configuración es solo un ejemplo, y la red podría ampliarse incluyendo más organizaciones, peers adicionales o distintos canales para gestionar diferentes votaciones.

FACULTAD DE INFORMÁTICA

4. Otra Documentación Relacionada

4.1. Hardware construido

Al ser este proyecto mayoritariamente de Software, de lo fisico solo se puede mostrar la Raspberry Pi 3 con cámara USB y pantalla LCD:



Figura 7: Materiales utilizados para la lectura del código QR.

La **Figura 7** muestra cómo se conecta tanto la cámara USB como la pantalla LCD, siguiendo el esquema de conexión presentado en la **Figura 2**. Este hardware se encuentra en la mesa de entrada.



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

4.2. Interfaz de Escaneo de QR

La interfaz web es presentada en el dispositivo móvil del votante. En la **Figura 8** se pueden observar la interfaz de inicio de sesión:



Figura 8: Interfaz Web del inicio de sesión de SmartPoll.

Donde se puede destacar que el usuario puede hacerlo a partir de su correo electrónico o DNI, ambos únicos en el sistema, y su contraseña de más de 4 caracteres.



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

Una vez que el usuario inicia la sesión, se muestra en pantalla el QR-Pase, como se puede ver en la **Figura 9**.



Figura 9: Interfaz Web QR-Pase.

Cuando el código es presentado frente a la cámara de la Raspberry Pi, el sistema lo valida y, una vez verificado, se muestra la siguiente interfaz de la **Figura 10**.



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

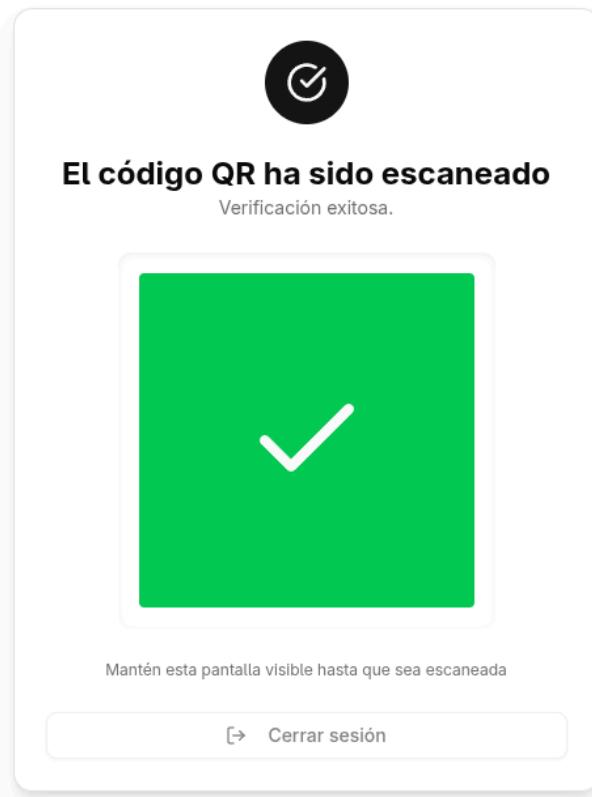


Figura 10: Interfaz Web QR-Pase escaneado con éxito.

En esta se observa que se le indica al usuario que ya puede proceder a emitir su voto.



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

4.3. Interfaz de votación

En la pantalla táctil de la Raspberry Pi del cuarto oscuro, se presenta la interfaz de votación la cual muestra una pantalla de espera, que indica que todavía no hay un usuario que haya escaneado el QR-Pase para votar, como se muestra en la **Figura 11**.



Figura 11: Interfaz de votación de SmartPoll en estado de espera.

Una vez que el usuario escanea con éxito el QR-Pase, la Raspberry Pi de la mesa de entrada se comunica con la de votación, pasando a la pantalla donde el usuario podrá seleccionar al candidato y votar como se presenta en la **Figura 12**.

FACULTAD DE INFORMÁTICA

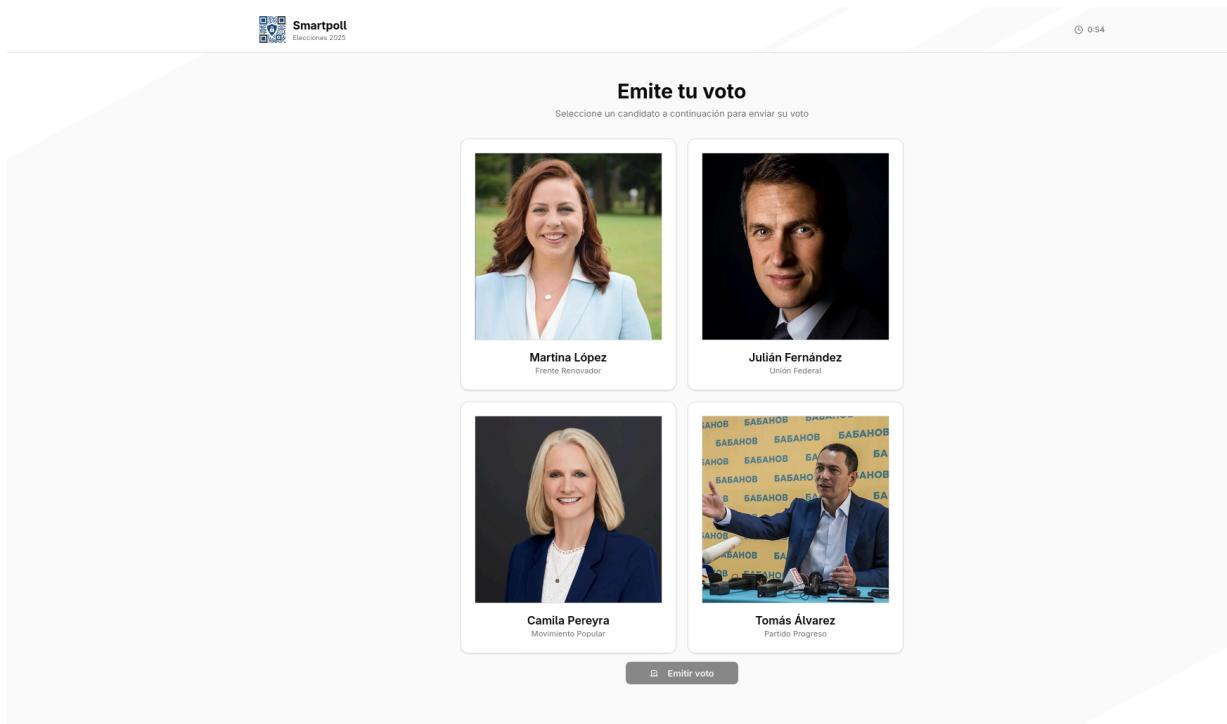


Figura 12: Interfaz de votación de SmartPoll con los candidatos.

4.4. Interfaz de auditoría

En la **Figura 13** se muestra la interfaz de auditoría, donde es posible visualizar la cantidad de votos que obtuvo cada candidato, junto con los porcentajes correspondientes. Esta interfaz se habilita únicamente una vez que la votación ha finalizado.



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

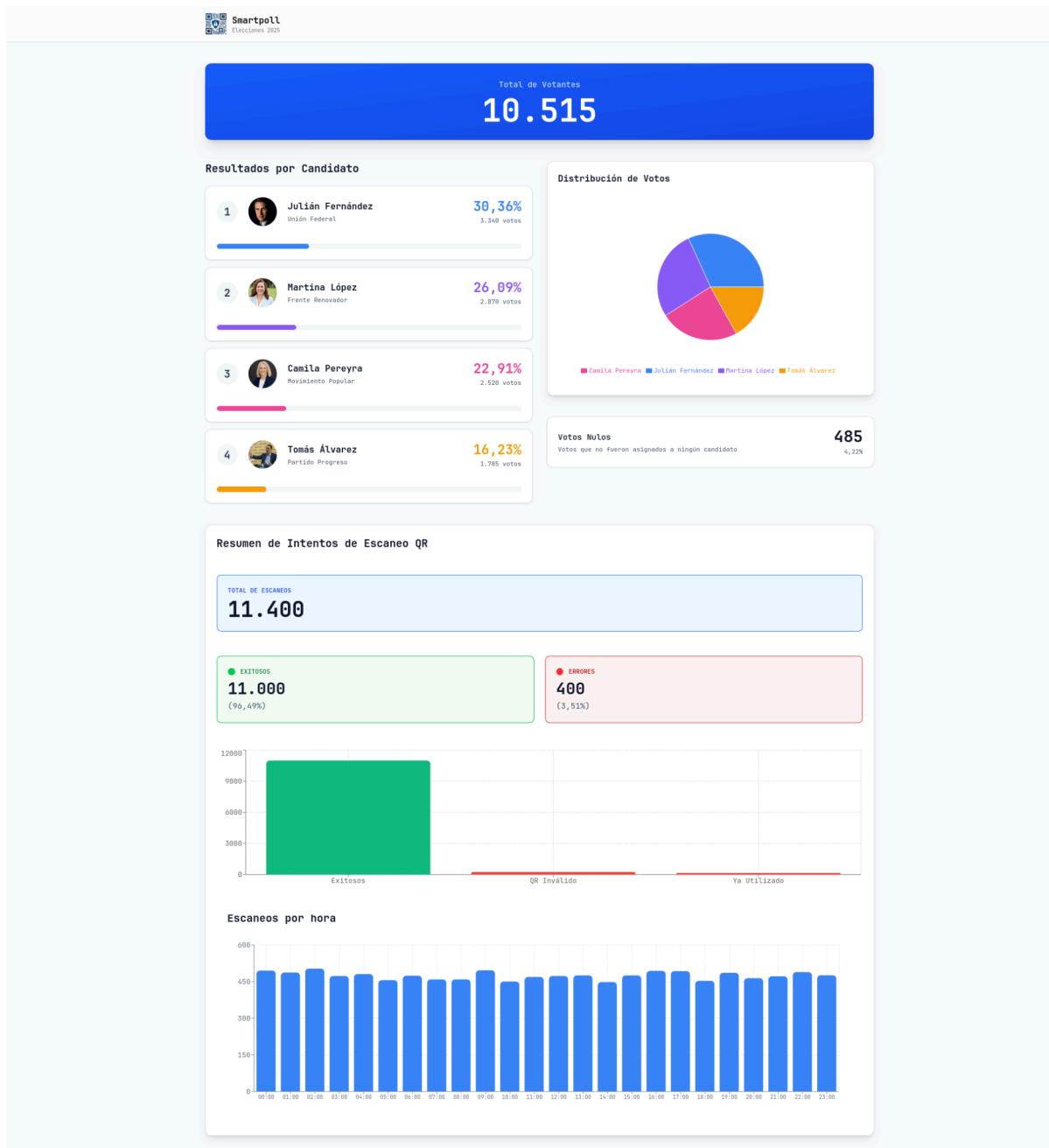


Figura 13: Interfaz de auditoría de SmartPoll.

Además, permite consultar los intentos de escaneo del código QR, diferenciando entre intentos correctos e incorrectos. También ofrece una visualización de cómo se distribuyeron estos intentos a lo largo del tiempo, agrupados por hora.

FACULTAD DE INFORMÁTICA

4.5. Interfaz de visualización de la Blockchain

En la **Figura 14** se observa la interfaz utilizada para visualizar el estado de la blockchain durante la ejecución del sistema. Esta interfaz corresponde a *Hyperledger Explorer*, una herramienta que permite monitorear en tiempo real los bloques, transacciones y nodos de la red blockchain.

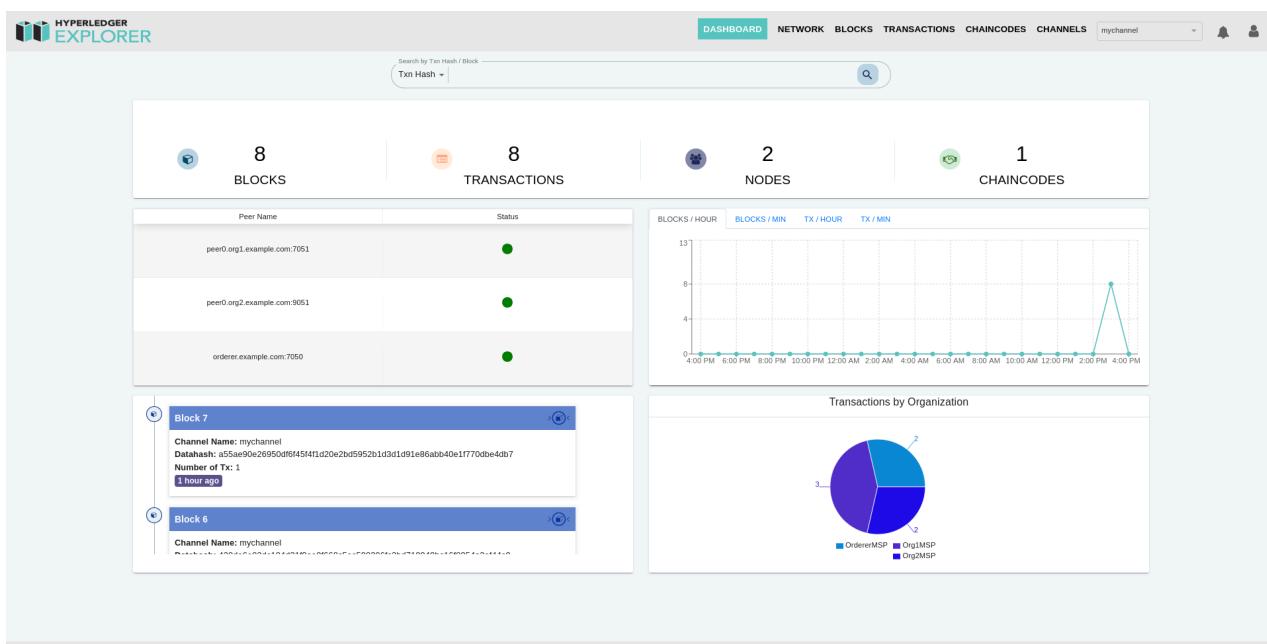


Figura 14: Interfaz de visualización de la blockchain.

La interfaz presenta un panel principal donde se resumen los elementos clave de la red, incluyendo la cantidad total de bloques, transacciones registradas, nodos activos y chaincodes desplegados. Además, se muestran los *peers* conectados junto con su estado, permitiendo verificar rápidamente la disponibilidad de cada nodo.

A la derecha se visualiza un gráfico que refleja la tasa de generación de bloques y transacciones a lo largo del tiempo, facilitando el seguimiento del comportamiento de la red. También se incluye un gráfico de distribución que detalla la cantidad de transacciones por organización, permitiendo identificar la participación de cada entidad dentro del canal.

En la parte inferior se puede navegar por los bloques individuales. Cada bloque detalla información relevante como el hash del bloque, el número de transacciones y el canal



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

asociado, lo cual permite inspeccionar en detalle la trazabilidad e integridad de los datos registrados en la blockchain.

Link al código fuente: <https://github.com/tpII/2025-G3-SmartPoll>

Link al video de presentación: <https://youtu.be/ENbJnc9eZks>

Link a la bitácora: <https://github.com/tpII/2025-G3-SmartPoll/wiki/Bit%C3%A1cora>

Funcionamiento de SmartPoll: <https://youtu.be/WeLFkxHDGlk>



FACULTAD DE INFORMÁTICA

Apéndice A: Materiales y Presupuesto

En la **Tabla A.1**, se presentan las listas de materiales que serán utilizados para llevar a cabo el proyecto, en ella se detallan: la cantidad, su ubicación y costo unitario.

Componente	Cantidad	Ubicación / Uso	Costo / U (USD)
Raspberry Pi 3 Model B	2	Mesa de ingreso y cuarto oscuro	\$72.00
Display Pantalla Lcd 16x2 Con I2c Incorporado	1	Mesa de ingreso	\$6.00
Cámara USB Logitech C170	1	Mesa de ingreso (lectura de QR-Pase)	\$36.00
Pantalla táctil para Raspberry Pi	1	Cuarto oscuro	\$42.00

Tabla A.1: Materiales utilizados en SmartPoll.

Todos los materiales que se observan en la **Tabla A.1** ya se encuentran disponibles para ser utilizados en el proyecto. Si bien para el proyecto se requiere de dos Raspberry Pi, solo se dispone de una sola puesto que la interfaz de votación se realizará con una notebook. No se requerirá hardware adicional con fondos de la cátedra.

Además fue necesario contratar servicios en la nube para alojar distintas aplicaciones desarrolladas. La mayoría de servicios utilizados están cubiertos por el *Free-Tier* de Amazon Web Services (AWS), por lo que no representan un costo adicional. Solo poseemos un costo bajo demanda de los siguientes recursos:

- **IPv4 pública:** \$0.005 USD por hora.
- **Instancia t4g.micro de EC2:** \$0.0084 USD por hora.