PRUEBA TÉCNICA
PRIMERA INSTANCIA - ANÁLISIS Y DISEÑO

Objetivo: Evaluar capacidades de análisis y diseño de sistemas.

Entregables requeridos:
- Diagrama de Casos de Uso
- Casos de Uso Extendidos (especificaciones detalladas)
- Modelo Relacional de Base de Datos
- Diagrama de Clases (según corresponda)
- Propuesta de Arquitectura de Software con justificación técnica

Autor: CZAJKOWSKI, Mirko Iván.

Destinatario: Cámara de Representantes de la Provincia de Misiones.

ÍNDICE

Diagrama de Casos de Uso

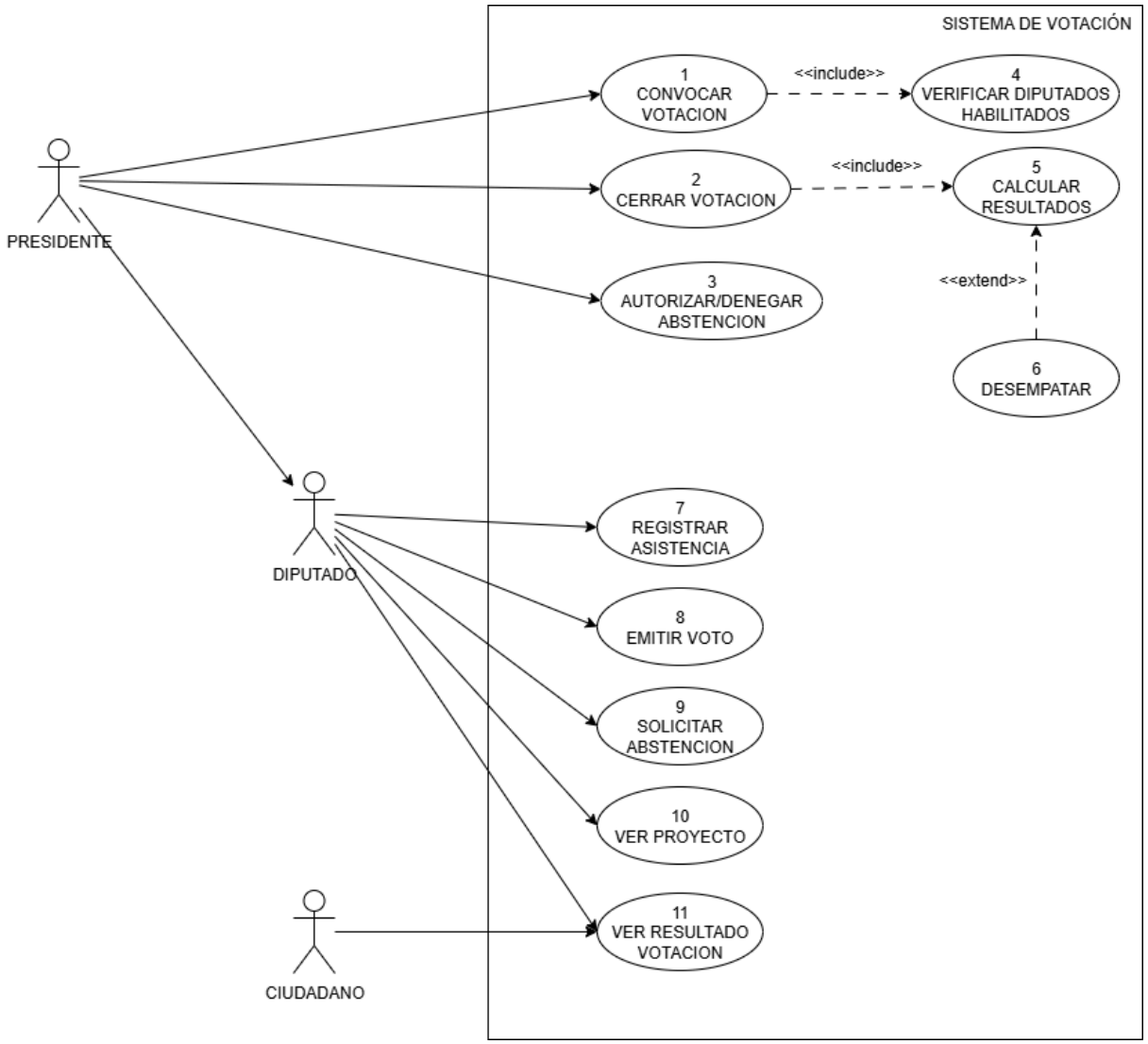

Imagen adjunta en la entrega para poder visualizar mejor su contenido.

Casos de Uso Extendidos

Caso de Uso: Convocar Votación	**ID:** 1	**Prioridad:** Alta
Actor Principal: Presidente		
Descripción breve: El Presidente de la sesión convoca y habilita la votación del proyecto.		
Pre condición: El Presidente de la sesión debe haber ingresado en el sistema con su usuario.		
Post condición: Se encuentra habilitada la votación del proyecto para los Diputados.		
Inicio: El Actor Presidente solicita al sistema Convocar a una Votación.		

Caso de Uso: Convocar Votación	**ID:** 1	**Prioridad:** Alta
Relaciones: ● Include: [ID 4] Verificar Diputados Habilitados.		
Flujo de eventos normales: 1. Actor Presidente: Solicita al sistema Convocar a una Votación. 2. Sistema: Verifica a los Diputados que tengan permitido votar. **[ID 4] Verificar Diputados Habilitados** 3. Sistema: Habilita la votación para los Diputados permitidos.		
Flujo de eventos alternativos: -		
Flujos de eventos de error: 2. Sistema: Verifica a los Diputados que tengan permitido votar y los diputados permitidos no superan el QUÓRUM necesario (mayoría de diputados habilitados sobre diputados activos) para Convocar a Votación. **[ID 4] Verificar Diputados Habilitados** 3. Sistema: Informa al Actor que no existe la cantidad necesaria (QUÓRUM) de Diputados habilitados para votar.		

Caso de Uso: Cerrar Votación	**ID:** 2	**Prioridad:** Alta
Actor Principal: Presidente		
Descripción breve: El Presidente de la sesión cierra la votación del proyecto para determinar su aprobación o no.		
Pre condición: Estar habilitada y en curso la votación del proyecto.		
Post condición: Se presentan los resultados de la votación.		
Inicio: El Actor Presidente solicita al sistema Cerrar la Votación.		
Relaciones: ● Include: [ID 5] Calcular Resultados.		
Flujo de eventos normales: 1. Actor Presidente: Solicita al sistema Cerrar la Votación. 2. Sistema: Calcula los resultados de la votación en curso. **[ID 5] Calcular Resultados**. 3. Sistema: Presenta los resultados de la votación.		
Flujo de eventos alternativos:		
Flujos de eventos de error:		

Caso de Uso: Autorizar/Denegar Abstención	**ID:** 3	**Prioridad:** Alta
Actor Principal: Presidente		

Caso de Uso: Autorizar/Denegar Abstención	**ID:** 3	**Prioridad:** Alta
Descripción breve: El Presidente de la sesión autoriza o deniega una solicitud de abstención de voto de un Diputado.		
Pre condición: Existir una solicitud de abstención de voto por parte de un Diputado. El Presidente de la sesión debe haber ingresado en el sistema con su usuario.		
Post condición: El Diputado queda autorizado o denegado para abstenerse al voto del proyecto.		
Inicio: El Actor Presidente solicita al sistema autorizar/denegar una solicitud de abstención.		
Relaciones: -		
Flujo de eventos normales: 1. Actor Presidente: Solicita al sistema autorizar/denegar una solicitud de abstención. 2. Sistema: Presenta al Actor las solicitudes de abstención. 3. Actor Presidente: Selecciona la solicitud que desea autorizar/denegar. 4. Actor Presidente: Autoriza/Deniega la solicitud de abstención del Diputado. 5. Sistema: Registra la autorización/denegación de abstención e informa al Diputado correspondiente.		
Flujo de eventos alternativos: -		
Flujos de eventos de error: -		

Caso de Uso: Verificar Diputados Habilitados	**ID:** 4	**Prioridad:** Alta
Actor Principal: Presidente		
Descripción breve: El Sistema verifica y retorna los diputados habilitados a votar el proyecto en base a la asistencia, Diputados activos en el sistema y abstenciones.		
Pre condición: El Presidente **[ID 1] Convocó una Votación**.		
Post condición: Se lista los Diputados habilitados para votar el proyecto actual y los Diputados abstenidos.		
Inicio: El Sistema recepciona una solicitud para verificar los diputados habilitados.		
Relaciones: -		
Flujo de eventos normales: 1. Sistema: Recepciona una solicitud para verificar los diputados habilitados. 2. Sistema: Recorre el registro de cada Diputado y controla su asistencia, su estado Activo y si posee una abstención autorizada/denegada. 3. Sistema: Presenta y retorna los Diputados habilitados para votar el proyecto actual, y los presentes abstenidos.		
Flujo de eventos alternativos: -		

Caso de Uso: Verificar Diputados Habilitados	**ID:** 4	**Prioridad:** Alta
Flujos de eventos de error: -		

Caso de Uso: Calcular Resultados	**ID:** 5	**Prioridad:** Alta
Actor Principal: Presidente		
Descripción breve: Se calculan los resultados de la votación y se define la aprobación o no del proyecto. Se detalla la cantidad de votos aprobados, la cantidad de votos reprobados, los diputados que se abstuvieron y los diputados que no votaron, el porcentaje en base al total de los votos aprobados, y los votos reprobados.		
Pre condición: El Presidente **[ID 2] Cerró una Votación**.		
Post condición: Se listan los resultados de la votación.		
Inicio: El Sistema recepciona una solicitud para calcular los resultados de la votación.		
Relaciones: ● Extend: **[ID 6] Desempatar**.		
Flujo de eventos normales: 1. Sistema: Recepciona una solicitud para calcular los resultados de la votación. 2. Sistema: Obtiene los votos emitidos y calcula el resultado comparando con los Diputados participantes de la votación. 3. Sistema: Presenta y retorna los resultados (mencionados en la descripción).		
Flujo de eventos alternativos: 2. Sistema: Obtiene los votos emitidos y calcula el resultado comparando con los Diputados participantes de la votación, dando como resultado un empate entre votos aprobados y reprobados. 3. Sistema: Procede a realizar un desempate en la votación. **[ID 6] Desempatar**. 4. Sistema: Presenta y retorna los resultados (mencionados en la descripción) aclarando que su definición fue mediante el “voto decisorio del Presidente”.		
Flujos de eventos de error: -		

Caso de Uso: Desempatar	**ID:** 6	**Prioridad:** Alta
Actor Principal: Presidente		
Descripción breve: El Presidente emite “el voto decisorio” para definir una votación empatada.		
Pre condición: El Sistema **[ID 5] Calculó los Resultados** de una votación y resultó empate.		
Post condición: La votación queda desempatada.		
Inicio: El Sistema recepciona una solicitud para desempatar una votación.		

Caso de Uso: Desempatar	**ID:** 6	**Prioridad:** Alta
Relaciones: -		
Flujo de eventos normales: 1. Sistema: Recepciona una solicitud para desempatar una votación. 2. Sistema: Presenta al Actor Presidente las opciones de Aprobar y Reprobar el proyecto para desempatar. 3. Actor Presidente: Selecciona la opción de Aprobar/Reprobar el proyecto. 4. Sistema: Retorna el resultado de la votación en base a la opción elegida por el Presidente.		
Flujo de eventos alternativos: -		
Flujos de eventos de error: -		

Caso de Uso: Registrar Asistencia	**ID:** 7	**Prioridad:** Alta
Actor Principal: Diputado		
Descripción breve: El Diputado registra su asistencia a la sesión.		
Pre condición: Haber una sesión abierta en la Cámara de Representantes.		
Post condición: El Diputado se registra como presente en la sesión.		
Inicio: El Actor Diputado solicita registrar su asistencia en la sesión.		
Relaciones: -		
Flujo de eventos normales: 1. Actor Diputado: Solicita registrar su asistencia en la sesión. 2. Sistema: Solicita al Actor ingresar con sus credenciales. 3. Actor Diputado: Ingresa sus credenciales. 4. Sistema: Verifica el registro del Diputado. 5. Sistema: Registra la asistencia como Presente del Diputado en la sesión.		
Flujo de eventos alternativos: -		
Flujos de eventos de error: 3. Actor Diputado: Ingresa sus credenciales. 4. Sistema: Verifica el registro del Diputado y no encuentra coincidencia con un usuario activo. 5. Sistema: Informa que las credenciales son incorrectas al Actor y regresa al paso número 2.		

Caso de Uso: Emitir Voto	**ID:** 8	**Prioridad:** Alta
Actor Principal: Diputado		
Descripción breve: El Diputado emite su voto para aprobar o reprobar el proyecto		

Caso de Uso: Emitir Voto	**ID:** 8	**Prioridad:** Alta
tratado.		
Pre condición: Estar habilitada y en curso la votación del proyecto. El Diputado debe haber registrado su asistencia a la sesión y estar habilitado para votar.		
Post condición: Queda registrado el voto de aprobación/reprobación del Diputado.		
Inicio: El Actor Diputado solicita Emitir su Voto.		
Relaciones: -		
Flujo de eventos normales: 1. Actor Diputado: Solicita Emitir su Voto. 2. Sistema: Presenta al Actor las opciones de Aprobar y Desaprobar el proyecto. 3. Actor Diputado: Selecciona la opción de Aprobar/Desaprobar el proyecto. 4. Sistema: Registra la opción seleccionada por el Actor.		
Flujo de eventos alternativos: -		
Flujos de eventos de error: 1. Actor Diputado: Solicita Emitir su Voto. 2. Sistema: Presenta al Actor las opciones de Aprobar y Desaprobar el proyecto. 3. Sistema: Informa que venció el tiempo de votación o que el Presidente ya Cerró la Votación del proyecto.		

Caso de Uso: Solicitar Abstención	**ID:** 9	**Prioridad:** Alta
Actor Principal: Diputado		
Descripción breve: El Diputado eleva su solicitud de abstención de votación del proyecto al Presidente de la sesión.		
Pre condición: El Diputado debe haber registrado su asistencia a la sesión; no debe haber votación en curso del proyecto.		
Post condición: La solicitud de abstención queda registrada en el sistema y elevada al Presidente de la sesión.		
Inicio: El Actor Diputado solicita abstención de votación al proyecto.		
Relaciones: -		
Flujo de eventos normales: 1. Actor Diputado: Solicita abstención de votación al proyecto. 2. Sistema: Presenta al Actor el título del Proyecto al cual solicitará abstención de voto y pregunta si realmente desea pedir la abstención. 3. Actor Diputado: Confirma que desea abstenerse del voto al proyecto mencionado en pantalla. 4. Sistema: Solicita al Actor el motivo por cual realiza la solicitud. 5. Actor Diputado: Ingresa el motivo de abstención. 6. Sistema: Registra la solicitud de abstención y lo notifica al Presidente de la sesión.		

Caso de Uso: Solicitar Abstención	**ID:** 9	**Prioridad:** Alta
Flujo de eventos alternativos: 2. Sistema: Presenta al Actor el título del Proyecto al cual solicitará abstención de voto y pregunta si realmente desea pedir la abstención. 3. Actor Diputado: Ingresa que no quiere realizar la solicitud de abstención. 4. Sistema: Regresa al Menú Principal (luego de haber registrado la asistencia). 4. Sistema: Solicita al Actor el motivo por cual realiza la solicitud. 5. Actor Diputado: No ingresa motivo de solicitud y sale de la opción de solicitud de abstención. 6. Sistema: Regresa al Menú Principal (luego de haber registrado la asistencia).		
Flujos de eventos de error: -		

Caso de Uso: Ver Proyecto	**ID:** 10	**Prioridad:** Baja
Actor Principal: Diputado		
Descripción breve: Se presenta al Diputado un resumen breve del proyecto próximo a votar.		
Pre condición: El Diputado debe haber registrado su asistencia a la sesión; debe haber un proyecto próximo a votar; no debe haber votación en curso del proyecto.		
Post condición: El resumen breve del proyecto próximo a votar se muestra al Diputado.		
Inicio: El Actor Diputado solicita ver el proyecto próximo a votar.		
Relaciones: -		
Flujo de eventos normales: 1. Actor Diputado: Solicita ver el proyecto próximo a votar. 2. Sistema: Recupera la información del proyecto y muestra el resumen breve en pantalla.		
Flujo de eventos alternativos: -		
Flujos de eventos de error: -		

Caso de Uso: Ver Resultado Votación	**ID:** 11	**Prioridad:** Alta
Actor Principal: Ciudadano / Diputado		
Descripción breve: Se presenta al Actor los resultados de una determinada votación de un proyecto.		
Pre condición: Debe haber al menos una Votación Finalizada con sus resultados disponibles.		
Post condición: Se presentan los resultados de la votación solicitada al Actor.		

Caso de Uso: Ver Resultado Votación	**ID:** 11	**Prioridad:** Alta
Inicio: El Actor solicita ver los resultados de una votación.		
Relaciones: -		
Flujo de eventos normales: 3. Actor: Solicita ver los resultados de una votación. 4. Sistema: Recupera los títulos de las votaciones finalizadas y las presenta al usuario para que elija una. 5. Actor: Elige un Proyecto que ya se haya votado. 6. Sistema: Presenta al usuario los resultados de la votación al Actor.		
Flujo de eventos alternativos: -		
Flujos de eventos de error: -		

Modelo Relacional de Base de Datos

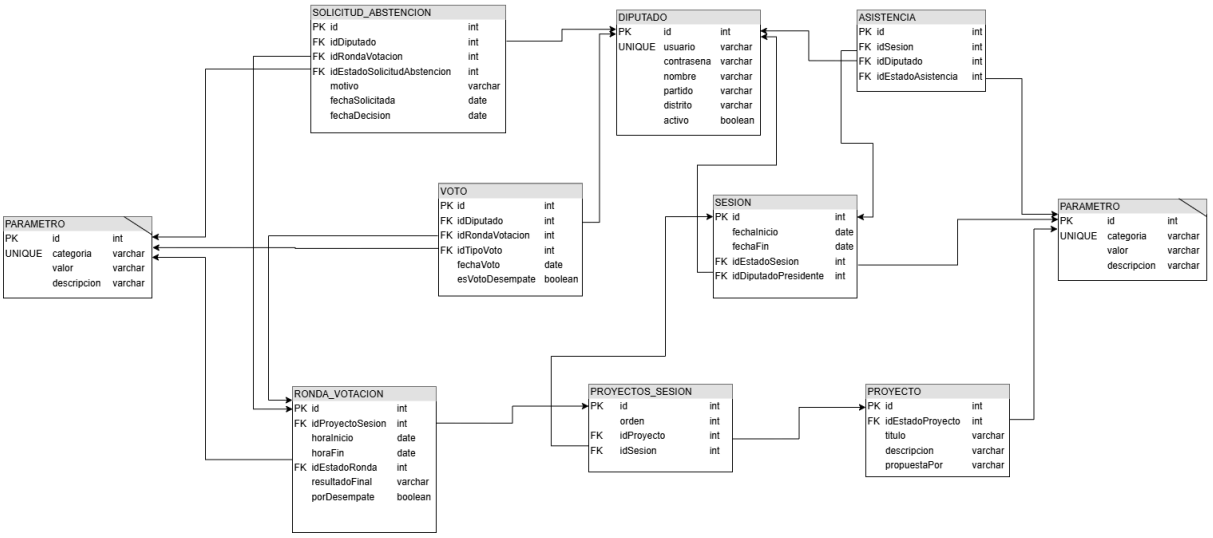

Imagen adjunta en la entrega para poder visualizar mejor su contenido.
Nótese que la tabla PARAMETRO se muestra por duplicado para facilitar su interpretación, así como algunas flechas que no apuntan directamente a la PK para evitar superposiciones por demás.

Diagrama de Clases

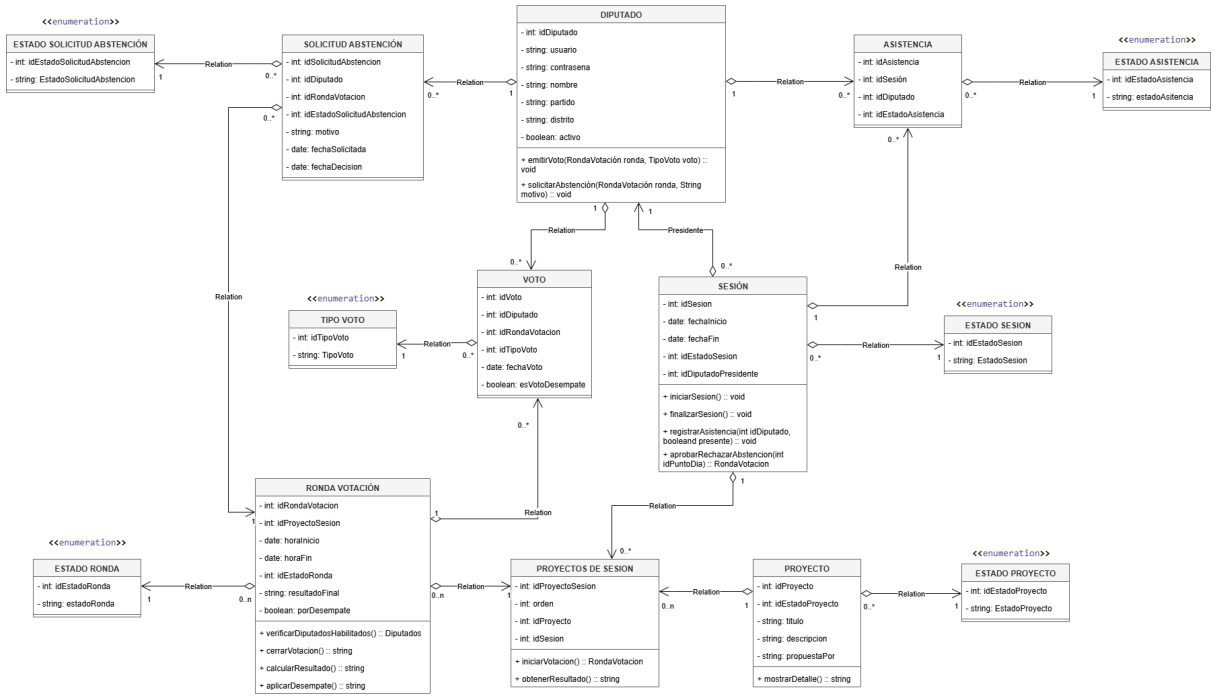

Imagen adjunta en la entrega para poder visualizar mejor su contenido.

Propuesta de Arquitectura de Software

Se propone una **Arquitectura de 3 Capas** como solución más adecuada y robusta. Esta arquitectura separa la aplicación en capas lógicas y físicas distintas, lo que mejora la mantenibilidad, escalabilidad y seguridad del sistema.

Propuesta: Arquitectura de 3 Capas

Esta arquitectura divide el sistema en las siguientes capas:

1. **Capa de Presentación (Frontend):** La interfaz con la que interactúan los usuarios (Presidente, Diputados y Ciudadanos).
2. **Capa de Lógica de Negocio (Backend):** Donde se procesan todas las reglas y operaciones.
3. **Capa de Persistencia de Datos (Base de Datos):** El sistema de almacenamiento donde reside toda la información.

Claro, aquí tienes la propuesta de arquitectura original, actualizada con la pila tecnológica que prefieres y las aclaraciones sobre seguridad.

1. Capa de Presentación (Frontend)

Interfaz de usuario que será como una aplicación web tradicional, enfocada en la eficiencia y la claridad.

- **Tecnología:** HTML5, CSS3 y JavaScript.

- **Justificación Técnica:**
    - **Universalidad y Simplicidad:** Son la base de toda la web, por lo que garantiza que la aplicación funcionará en cualquier navegador moderno sin necesidad de dependencias complejas, ideal para el entorno controlado de la Cámara de Representantes.
    - **Dinamismo con AJAX:** Con JavaScript se utilizarán peticiones AJAX para comunicarse con el backend. Esto permite actualizar partes de la página, como el estado de la votación o los resultados, sin tener que recargarla por completo, cumpliendo el objetivo de optimizar los tiempos y mejorar la transparencia.
    - **Bajo Acoplamiento:** Esta capa se encarga únicamente de mostrar la información y capturar las interacciones del usuario. Toda la lógica compleja se encuentra en el backend, lo que facilita futuras actualizaciones visuales.

2. Capa de Lógica de Negocio (Backend)

Servidor que contendrá toda la lógica de la aplicación. Se expondrá al frontend a través de una API RESTful, usando JSON como formato de intercambio de datos.

- **Tecnología Recomendada:** PHP 8+ utilizando el framework Laravel.
- **Justificación Técnica:**
    - **Desarrollo Robusto y Organizado:** Laravel es un framework de PHP moderno que organiza el código bajo el patrón Modelo Vista Controlador (MVC). Esto hace que la aplicación sea mucho más fácil de mantener y escalar en el futuro.
    - **Seguridad Integrada:** Laravel proporciona de serie mecanismos de protección contra las vulnerabilidades más comunes. Su ORM, Eloquent, utiliza sentencias preparadas que previenen de forma nativa los ataques de inyección SQL. Su motor de plantillas, Blade, escapa automáticamente las salidas para prevenir ataques XSS.
    - **Ecosistema Completo:** Ofrece soluciones integradas para gestionar rutas de la API, autenticación de usuarios (Diputados, Presidente) y autorización basada en roles (con opciones limitadas para los Ciudadanos que deseen ver los resultados), lo cual es fundamental para este sistema.

3. Capa de Persistencia de Datos (Base de Datos)

Capa que almacena y recupera los datos de forma segura y consistente, basándose en el modelo relacional propuesto anteriormente.

- **Tecnología Recomendada:** MySQL.
- **Justificación Técnica:**
    - **Integridad Relacional:** Un sistema de base de datos relacional se adapta para este modelo, ya que las restricciones de clave primaria y foránea garantizan la consistencia de los datos, asegurando que cada voto corresponda a un diputado y a una ronda de votación válida.
    - **Transacciones Seguras (ACID):** Las votaciones son operaciones atómicas. Las propiedades ACID de estas bases de datos garantizan que una ronda de

votación se registre por completo o no se registre nada, evitando datos corruptos o resultados inconsistentes.

- **Compatibilidad:** MySQL se integra de manera nativa y eficiente con PHP y Laravel, lo que simplifica y asegura la comunicación entre la capa de lógica y la de datos.

Otras Características

En cuanto a la seguridad del sistema, no es una capa adicional, pero es una responsabilidad transversal que se gestiona de la siguiente manera:

- **Comunicación Segura:** Todo el tráfico entre el frontend y el backend debe ser encriptado obligatoriamente a través del protocolo HTTPS.
- **Prevenciones:** En la capa de backend se maneja mediante las características de Laravel para evitar peligros como inyecciones SQL y XSS.

Para desplegar el proyecto:

- **Despliegue:** Se utilizarán contenedores (Docker) para empaquetar la aplicación y sus dependencias. Esto asegura que el sistema funcione de manera consistente en cualquier entorno y simplifica el despliegue y la escalabilidad.