

Diseño de un sistema de información para el seguimiento de los resultados de preconteo electoral, de las elecciones regionales en candidatos a la gobernación en el departamento de Casanare SISPRECAS-Soft.

John Walter Guerrero León

Universidad Nacional Abierta y a Distancia UNAD
Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI
Ingeniería de Sistemas

2024

Diseño de un sistema de información para el seguimiento de los resultados de preconteo electoral, de las elecciones regionales en candidatos a la gobernación en el departamento de Casanare SISPRECAS-Soft.

Estudiante:

John Walter Guerrero León

Proyecto aplicado como requisito parcial para optar el título de:

Ingeniero de Sistemas

Director de proyecto de grado:

Ing. Angela Dayan Garay Villada

Universidad Nacional Abierta y a Distancia – UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería – ECBTI

Ingeniería de Sistemas

2024

Agradecimientos

A Dios por regalarme la vida y poner hermosas personas durante toda mi etapa académica, las cuales me abrieron grandes puertas de bendiciones. También quiero agradecer a mi familia, a mi madre por llevarme en sus oraciones, a mis hermanos quienes han puesto granitos de arena apoyándome en cada una de mis etapas de desarrollo personal, a mi esposa Pilar Albarracín quien me fundamento en la importancia de terminar el pregrado y me dio ánimo para culminar.

Agradezco también a mi amigo Luis Sánchez, con quien comparto el gusto por la programación y llevamos años luchando día a día por alcanzar nuestros sueños.

A la Universidad Nacional Abierta y a Distancia - UNAD, por darme la oportunidad de ser uno de sus estudiantes y brindarme una educación de calidad, la cual me permitió crecer intelectualmente y desarrollar toda mi etapa académica, teniendo en cuenta mi condición de estudiante y empleado. Agradezco infinitamente todas las herramientas, recursos y conocimientos que me brindó para la realización de mi carrera de pregrado. A los docentes de cada una de las asignaturas cursadas, especialmente a mi director de programa académico el Ingeniero Rafael Pérez y mi directora de tesis la Ing. Angela Garay, por su orientación durante el desarrollo de este proyecto. Su experiencia, conocimiento y dedicación fueron fundamentales para mi aprendizaje y mi crecimiento personal, porque siempre estuvo con la buena disposición para orientarme y proponer soluciones a cada pregunta, cada duda. Lo cual me permitió adquirir los conocimientos suficientes para desarrollar este proyecto.

A todos,

¡Infinitas gracias

Resumen

Teniendo como base los resultados del análisis presentado por la Misión de Observación Electoral (MOE), quienes afirman la alta posibilidad de presentarse riesgo de fraude electoral en las elecciones regionales del 29 de octubre de 2023 en Casanare (MOE 2023). Se decide diseñar un sistema de información para seguimiento a los resultados de preconteo electoral en el departamento, específicamente para los candidatos a la gobernación, ya que es el cargo más alto para ocuparse como primera autoridad administrativa del departamento. La implementación del sistema se ejecutó a través de la arquitectura cliente/servidor, la cual permite dar órdenes o generar peticiones desde un dispositivo con acceso a internet (cliente) para posteriormente el servidor responder a las solicitudes. Este modelo es ideal porque permite a los usuarios conectados a la aplicación registrar la información expedida por los jurados de votación para cada una de las mesas instaladas en el departamento. A través de un smartphone o teléfono inteligente con acceso a internet, lo que permite posteriormente obtener los resultados o la votación por cada opción registrada en el tarjetón electoral (candidatos, votos en blanco, votos no marcados, votos nulos), información que puede ser consultada ya sea por cada mesa de votación, por cada puesto de votación o por cada municipio. La intención de este proyecto es contribuir a disminuir los índices de riesgo electoral en el departamento, además de conocer el candidato electo como gobernador y comparar la información publicada por la Registraduría Nacional, con la información suministrada en nuestro sistema; Teniendo como única fuente el Formato E14, el cual expiden los jurados de votación en cada una de las mesas instaladas para las elecciones en el departamento.

Palabras clave: Fraude electoral, preconteo, elecciones, desarrollo de software

Abstract

Based on the results of the analysis presented by the Electoral Observation Mission (MOE), who affirm the high possibility of a risk of electoral fraud in the regional elections of October 29, 2023 in Casanare (MOE 2023). It was decided to design an information system to monitor the results of the electoral pre-count in the department, specifically for the candidates for governor, since it is the highest position to hold as the first administrative authority of the department. The implementation of the system was executed through the client/server architecture, which allows giving orders or generating requests from a device with Internet access (client) and then the server responds to the requests. This model is ideal because it allows users connected to the application to record the information issued by the voting juries for each of the tables installed in the department. Through a smartphone or smartphone with Internet access, which allows you to subsequently obtain the results or the vote for each option registered on the electoral card (candidates, blank votes, unmarked votes, null votes), information that can be consulted either by each voting station, by each voting station or by each municipality. The intention of this project is to contribute to reducing electoral risk rates in the department, in addition to knowing the candidate elected as governor and comparing the information published by the National Registry, with the information provided in our system; Having as the only source the E14 Format, which is issued by the voting juries at each of the tables installed for elections in the department.

Keywords: Electoral fraud, pre-count, elections, software development

Tabla de Contenido

Agradecimientos.....	3
Resumen	4
Abstract.....	5
Tabla de Contenido	6
Lista de Tablas.....	9
Lista de Figuras	10
Introducción	14
Planteamiento del problema	16
Formulación del problema	18
Justificación del problema	19
Objetivo General.....	21
Objetivos Específicos.....	22
Estado del Arte	23
Marco Teórico.....	26
Democracia y Transparencia Electoral.....	26
Ley 2108 del 29 de julio de 2021.	26
Participación Ciudadana y Fortalecimiento Institucional.....	26
Riesgo de Fraude Electoral.....	27
Tecnologías de Desarrollo	28
Metodología de prototipo	31
Requisitos del Sistema.....	31
Diseño del Prototipo.....	32
Desarrollo del Prototipo	32
Pruebas del Prototipo	32

Evaluación del Prototipo	33
Cronograma de actividades.....	34
Recursos para el desarrollo del proyecto.....	36
Hardware:	36
Recurso humano para el desarrollo del proyecto:.....	37
Recurso humano para la ejecución del proyecto.....	38
Recursos económicos.....	39
Diseño de la investigación	41
Análisis de los requerimientos del sistema.....	43
Encuesta en línea para la viabilidad del proyecto.....	47
Diseño gráfico para interpretar los requisitos del sistema.	50
Diagrama de clases:.....	51
Descripción de las clases:.....	52
Normalización de la base de datos.	55
Primera forma normal:	55
Segunda y tercera forma normal:.....	56
El proceso de la segunda y tercera forma normal se realizó para identificar los campos foráneos en tablas relacionas entre sí y eliminar duplicados de información. Generalmente una base de datos esta normalizada en su tercera forma normal.....	56
Diagramas de casos de uso:.....	57
Diseño de la interfaz de usuario (UI).....	64
Desarrollo del prototipo	68
Estructura de la aplicación	69
Manejo de operaciones CRUD en la aplicación	74

	8
Plan de pruebas.....	88
Resultados	92
Repositorio de la aplicación (Backend y FrontEnd).....	100
Glosario	101
Conclusiones.....	105
Referencias bibliográficas.....	106

Lista de Tablas

Tabla 1 <i>Recursos económicos.</i>	39
Tabla 2 <i>Descripción de la clase usuarios.</i>	52
Tabla 3 <i>Descripción de la clase mesas.</i>	53
Tabla 4 <i>Descripción de la clase puestos de votación.</i>	53
Tabla 5 <i>Descripción de la clase candidatos.</i>	53
Tabla 6 <i>Descripción de la clase partidos.</i>	54
Tabla 7 <i>Descripción de la clase municipios.</i>	54
Tabla 8 <i>Descripción de la clase zonas.</i>	55
Tabla 9 <i>Descripción del caso de uso, inicio de sesión.</i>	59
Tabla 10 <i>Descripción del caso de uso registro de información.</i>	61
Tabla 11 <i>Descripción del caso de uso consulta de información.</i>	63

Lista de Figuras

Figura 1 <i>Desarrollo de la fase 1 y 2 del desarrollo del proyecto.</i>	34
Figura 2 <i>Desarrollo de la fase 3 del desarrollo del proyecto.</i>	34
Figura 3 <i>Desarrollo de la fase 4 y 5 del desarrollo del proyecto.</i>	35
Figura 4 <i>Diagrama de flujo para identificar las posibles entidades en la base de datos</i>	50
Figura 5 <i>Descripción del diagrama de clases</i>	51
Figura 6 <i>Normalización de base de datos. primera forma normal.</i>	56
Figura 7 <i>Normalización de base de datos. Segunda y tercera forma normal</i>	57
Figura 8 <i>Diagrama de caso de uso actores</i>	58
Figura 9 <i>Diagrama de caso de inicio de sesión.</i>	58
Figura 10 <i>Caso de uso registro de información</i>	60
Figura 11 <i>Caso de uso, consulta de información.</i>	62
Figura 12 <i>Formulario de inicio de sesión.</i>	64
Figura 13 <i>Panel de administración de usuarios editores</i>	65
Figura 14 <i>Formulario E-14.</i>	66
Figura 15 <i>Panel de administración. Página de consultas</i>	67
Figura 16 <i>Panel de administración. Página de resultados</i>	67
Figura 17 <i>Estructura del proyecto</i>	70
Figura 18 <i>Script principal de la base de datos</i>	71
Figura 19 <i>Creación de la base de datos en Vercel</i>	71
Figura 20 <i>Ejecución del backend del proyecto backend en local.</i>	72
Figura 21 <i>Proyecto backend ejecutado en Vercel</i>	72

Figura 22 <i>FrontEnd ejecutado en local</i>	73
Figura 23 <i>Vista despliegue del FrontEnd en Vercel</i>	73
Figura 24 <i>Visualización del Frontend en https://proyecto-elecciones-frontend.vercel.app</i>	74
Figura 25 <i>Visualización del token guardado en el navegador, para un usuario autenticado en el proyecto SISPRECAS-soft</i>	75
Figura 26 <i>Controlador de usuarios, usado para manejar el CRUD del endpoint usuarios</i>	76
Figura 27 <i>Función del endpoint usuarios para listar los usuarios del sistema</i>	76
Figura 28 <i>Función del endpoint usuarios, para listar los datos de un usuario del sistema, se accede al hacer una petición HTTP de tipo GET a /usuarios/:id</i>	77
Figura 29 <i>Función del endpoint usuarios, para registrar un nuevo usuario. Se accede al hacer una petición HTTP de tipo POST a /usuarios</i>	77
Figura 30 <i>Función del endpoint usuarios, para actualizar un usuario del sistema. Se accede al hacer una petición HTTP PUT a /usuarios/:id</i>	78
Figura 31 <i>Función del endpoint usuarios, para eliminar un usuario del sistema. Se accede al hacer una petición HTTP DELETE a /usuarios/:id</i>	78
Figura 32 <i>Vista del menú usuarios disponible para el usuario administrador</i>	79
Figura 33 <i>Visualización del detalle de usuario desde el perfil del administrador</i>	80
Figura 34 <i>Visualización de la actualización de usuario desde el usuario administrador</i>	81
Figura 35 <i>Visualización de la edición de usuario desde el usuario administrador</i>	82
Figura 36 <i>Visualización de la edición de usuario desde el usuario administrador</i>	83

Figura 37 <i>Función del controlador consultas, dedicada a manejar las peticiones al endpoint/consultas/resultados/candidatos, la cual permite generar los datos consolidados de los candidatos, para la generación de informes.....</i>	84
Figura 38 <i>Función del controlador consultas, dedicada a manejar las peticiones al endpoint/consultas/resultados/otras-opciones, la cual permite generar los datos consolidados de votos en blanco, votos no marcados y nulos, para la generación de informes.</i>	85
Figura 39 <i>Visualización menú consultas para usuarios con roles administrador y candidatos</i>	86
Figura 40 <i>Dependencia "React PDF" en el archivo package.json del frontend realizado con React</i>	87
Figura 41 <i>Layout y estilos usados para el pdf a generar.....</i>	87
Figura 42 <i>Uso de la librería "React PDF" para generar el documento pdf con los datos consolidados</i>	88
Figura 43 <i>Validación de inicio de sesión.....</i>	89
Figura 44 <i>Validación de campos al enviar el formulario E14</i>	89
Figura 45 <i>Verificación de funcionamiento de la API con Postman.....</i>	90
Figura 46 <i>Verificación de restricciones de acceso a personas o sistemas no autorizados</i>	91
Figura 47 <i>Inicio de sesión.....</i>	92
Figura 48 <i>Página inicial del usuario Administrador.....</i>	93
Figura 49 <i>Añadir información de la mesa de votación con el formulario E14-1</i>	93
Figura 50 <i>Añadir información de la mesa de votación con el formulario E14-2.....</i>	94
Figura 51 <i>Página de Consulta con filtros por departamento, municipio, puesto de votación y mesas.....</i>	94

	13
Figura 52 <i>Página de resultados: Consolidado de votaciones</i>	95
Figura 53 <i>Generación de informes desde la página de consultas</i>	96
Figura 54 <i>Visualización del perfil de usuario</i>	97
Figura 55 <i>Página de gestión de usuarios</i>	97
Figura 56 <i>Agregar nuevo usuario</i>	98
Figura 57 <i>Visualización de perfil de usuario con opción de edición</i>	98
Figura 58 <i>Edición de usuario</i>	99

Introducción

En las elecciones territoriales celebradas el 29 de octubre de 2023 en Colombia, se hizo evidente una creciente demanda por herramientas tecnológicas que aseguren la integridad y transparencia en los resultados electorales. Estos resultados, vitales para la toma de decisiones democráticas, son inicialmente generados por los jurados de votación durante el proceso de preconteo electoral, llevado a cabo en cada mesa de votación en todo el país, al concluir la jornada electoral (Gran foro por la transparencia en las elecciones 2023 - Más Contenido – Eltiempo.com, s/f).

Con el propósito de garantizar la integridad de la voluntad de los votantes y ofrecer un seguimiento detallado de este proceso, se ha diseñado el sistema de información en línea denominado SISPRECAS-Soft. Esta innovadora solución permitirá supervisar minuciosamente el proceso de preconteo electoral en todas las mesas de votación distribuidas en los diferentes municipios y puestos de votación del departamento de Casanare.

El desarrollo de esta tecnología se llevó a cabo siguiendo la metodología de prototipado, como sugiere Martínez (2015), lo que permite partir de un modelo cercano al resultado final. Se emplearon lenguajes de programación ampliamente reconocidos, como JavaScript y React.js, junto con el formato de intercambio de datos JSON. Además, se utilizaron tecnologías del lado del servidor como Node.js y Express para garantizar un funcionamiento eficiente de la aplicación. Finalmente, para el almacenamiento seguro y organizado de los datos, se optó por el robusto sistema de gestión de bases de datos PostgreSQL.

El objetivo principal de este proceso integral, que va desde el análisis hasta la implementación del sistema SISPRECAS-Soft, es contribuir con una pequeña parte para asegurar la integridad de la voluntad de los votantes. En este contexto, exploraremos en detalle la funcionalidad del sistema de información y el impacto que esta herramienta tecnológica puede tener en los procesos electorales de los candidatos a la gobernación en el departamento de Casanare.

Planteamiento del problema

Colombia es un país democrático y sus líderes o representantes se eligen a partir de procesos electorales que se llevan a cabo en cada uno de los treinta y dos departamentos; incluyendo sus respectivos municipios, corregimientos y veredas. Estos procesos electorales deben ser justos para el funcionamiento efectivo de una democracia, ya que permiten que la voluntad de los ciudadanos se exprese de manera equitativa y que todos los votantes tengan la misma oportunidad de participar.

Actualmente Colombia es un país criticado y evaluado en sus procesos electorales. Un caso puntual que ilustra esta preocupación es el informe publicado por la Misión de Observación Electoral (MOE) el 28 de septiembre de 2023. En donde se indicó que existía riesgo de fraude electoral en las elecciones de autoridades locales a realizarse el 29 de octubre del mismo año. "596 municipios presentan algún nivel de riesgo. De estos, 317 se encuentran en riesgo medio, 212 en riesgo alto y 67 en riesgo extremo por variables de dominio electoral, participación electoral, votos nulos, tarjetones no marcados y trashumancia electoral" (*"166 municipios presentan algún nivel de riesgo por coincidencia de factores indicativos de fraude electoral y de violencia para las elecciones de autoridades locales 2023": MOE - MOE - Misión de Observación Electoral, s/f*).

Esta evaluación pone de manifiesto la urgente necesidad de abordar y resolver los problemas relacionados con la integridad de los procesos electorales en el país.

Una forma de velar por la integridad de la decisión de los votantes es la inclusión de nuevas tecnologías en los procesos electorales, lo que permite mejorar la eficiencia, accesibilidad y transparencia de las elecciones. Estas innovaciones pueden facilitar el registro de votantes, agilizar el proceso de votación y proporcionar mecanismos más seguros para la recopilación y tabulación de los resultados. Al incorporar tecnologías como la votación electrónica, sistemas de seguimiento en tiempo real o la verificación biométrica. Se pueden reducir las barreras a la participación ciudadana, aumentar la precisión de los resultados y fortalecer la integridad del proceso electoral (*¿Qué pasa con el voto electrónico en Colombia?*, s/f).

Formulación del problema

¿Cómo recopilar los datos del preconteo electoral en el departamento de Casanare, de tal forma que se pueda hacer seguimiento a los resultados expedidos por los jurados de votación y de este modo contribuir a disminuir el riesgo de fraude electoral, y se proteja la integridad de la decisión de los votantes?

Justificación del problema

Desde el año 2007 la Misión de Observación Electoral en Colombia (MOE) ha venido generando mapas de riesgos electorales a través de los datos históricos producidos por la Registraduría Nacional. Toda esta información es procesada por equipos de analistas independientes, universidades, centros de pensamiento y organizaciones sociales del país. Quienes han venido advirtiendo la existencia de factores indicativos de fraude en elecciones pasadas y manifestando la preocupación para elecciones futuras; como fue el caso de las elecciones de autoridades locales realizadas el 29 de octubre del año 2023 en donde se venía advirtiendo el riesgo de fraude (MOE) antes de las elecciones (*Mapas de Riesgo Electoral - MOE - Misión de Observación Electoral, s/f*).

Durante todos estos años se han realizado alrededor de 11 elecciones de autoridades locales, regionales y nacionales. Si bien es cierto, que, el riesgo de fraude electoral ha disminuido, el hecho que aún se continúe encontrando riesgos en más del 50% de los municipios del país (MOE 2023), es un asunto que no deja de importar y la necesidad de proteger la integridad de la decisión de los votantes es evidente.

La falta de control para monitorear inmediatamente los resultados del preconteo de votos durante las elecciones, hace que el traslado de mesas al lugar de escrutinio cree una oportunidad potencial para sabotaje o fraude y lamentablemente esta vulnerabilidad ha contribuido a que los procesos electorales sean un problema crítico en Colombia.

Todos estos cuestionamientos desaprueban la confianza en el proceso democrático y ponen en duda la integridad de las elecciones. En este contexto, es importante que las autoridades e instituciones académicas se comprometan en fortalecer la confiabilidad y transparencia de las elecciones. A través de programas y proyectos alineados a contrarrestar esta problemática y de este modo contribuir, que los resultados expedidos por los jurados de votación sean siempre los mismos resultados obtenidos en los escrutinios electorales. Y así finalmente, lograr que a través de estas prácticas los procesos electorales sean un poco más justos en el departamento de Casanare.

Objetivo General

Diseñar un sistema de información mediante la metodología de prototipo para el seguimiento de los resultados de preconteo electoral, el cual se probará con los resultados publicados por la Registraduría Nacional de las elecciones regionales para la gobernación del departamento de Casanare 2023.

Objetivos Específicos

Analizar los requerimientos del sistema de información para obtener los resultados del preconteo electoral.

Construir los diferentes módulos de trabajo que permitan el correcto funcionamiento del sistema.

Realizar el plan de pruebas de software para identificar posibles vulnerabilidades de seguridad en la información.

Estado del Arte

Título: InfoVotantes

Autores: Área de Gestión de Software.

Institución: Registraduría Nacional del Estado Civil Colombiano.

Publicación: 2022.

Resumen: La Registraduría Nacional del Estado Civil ha desarrollado la aplicación móvil Infovotantes como una herramienta tecnológica para que la ciudadanía acceda a información relevante sobre las elecciones. A través de esta aplicación, los ciudadanos pueden consultar detalles como su lugar de votación, las rutas para acceder al puesto de votación, si fueron designados como jurados de votación y, en caso afirmativo, cuándo y dónde recibirán capacitación para desempeñar ese rol. Además, la aplicación proporciona información sobre los candidatos a la Presidencia y Vicepresidencia, la tarjeta electoral y otros datos relevantes. Así mismo, el Registrador Nacional del Estado Civil, Alexander Vega Rocha, destacó el compromiso de la Registraduría con la modernización y la tecnología para facilitar la participación ciudadana en el proceso electoral (*Infovotantes, una herramienta tecnológica de la Registraduría Nacional al (...) - Registraduría Nacional del Estado Civil, s/f*).

Título: Software de escrutinio.

Autores: Área de Gestión de Software.

Institución: Registraduría Nacional del Estado Civil Colombiano.

Publicación: 2021.

Resumen: La Organización Electoral (Consejo Nacional Electoral y Registraduría Nacional del Estado Civil), adquirió un software de escrutinio nacional a finales de diciembre de 2021 para las elecciones de Congreso, presidente y vicepresidente de Colombia en 2022. El software fue adquirido mediante licitación pública y es operado por los magistrados del Consejo Nacional

Electoral. El software permite tener el control de los resultados electorales a nivel nacional. Dentro de las funciones que permite, está la de registrar la información suministrada por los jurados de votación en el formato E14. Este software es operado directamente por autoridades delegadas por el Consejo Nacional Electoral, quienes son los encargados de registrar o actualizar los votos obtenidos por cada candidato en cada mesa instalada en cada puesto de votación en todos los municipios del país. Además guarda e imprime informe de las personas que se registran o que ingresan información al sistema (*La Organización Electoral es propietaria del software de escrutinio para las (...) - Registraduría Nacional del Estado Civil, s/f*).

A sí mismo, en Colombia se han desarrollado diferentes proyectos relacionados al tema de las elecciones, algunos que son relevantes al tema son:

Título: Sistema de Voto Electrónico Seguro - SVES

Autores: Grupo de investigación - GIMATICA

Institución: Universidad de Cartagena

Publicación: 2014.

Resumen: El grupo de investigación – GIMATICA, de la Universidad de Cartagena desarrolló un sistema de votación electrónica en lenguaje de programación JAVA, con el objetivo de mejorar los procesos de consulta para ciertos cargos dentro de la institución. Anteriormente, estos procesos se realizaban manualmente, lo que implicaba ciertas limitaciones y dificultades. Reconociendo la necesidad de modernizar este proceso y aprovechar las ventajas del voto electrónico, se llevó a cabo un proyecto de investigación aplicada. El objetivo principal del

proyecto fue diseñar un sistema de votación electrónica seguro que cumpliera con medidas de seguridad lógicas, físicas y tecnológicas necesarias, así como implementar políticas de seguridad para su utilización. Para lograrlo, se llevó a cabo un estudio exhaustivo de sistemas de información, metodologías de desarrollo, procedimientos actuales y sistemas de votación electrónica. Además permitía la votación online de cada estudiante universitario y la respectiva generación de informes.

Título: Sistema de Información Urna Electrónica para Procesos Operacionales de Consulta Electoral de la Unidad Central del Valle del Cauca – UCEVA.

Autores: David Steven Duran Vallejo (estudiante).

Institución: Unidad Central del Valle del Cauca (UCEVA),

Publicación: 2018.

Resumen: En el contexto democrático de una sociedad, incluyendo organizaciones educativas como la Unidad Central del Valle del Cauca (UCEVA), David Steven Duran Vallejo (estudiante), implementó una solución tecnológica para mejorar la problemática de los procesos electorales manuales, que eran repetitivos y costosos en términos de recursos humanos y económicos. Para abordar esta situación, David Steven desarrolló una aplicación llamada SIUE (Sistema de Información Urna Electrónica), que ofrece una solución ágil y segura para gestionar los procesos electorales. Esta aplicación permite llevar a cabo elecciones como la elección de docentes ante el Consejo Directivo de la UCEVA, con la autenticación de votantes, emisión de votos, publicación de resultados y generación de informes en tiempo real. Además de reducir costos.

Marco Teórico

A continuación, se presentan las bases teóricas que se tomaron para el diseño de la aplicación:

Democracia y Transparencia Electoral.

La democracia es un pilar fundamental de cualquier sociedad y se basa en la participación de los ciudadanos en la toma de decisiones a través de elecciones libres y justas. La integridad de los procesos electorales es esencial para garantizar la representatividad y la confianza en el sistema democrático. Nuestro país Colombia está basado en un sistema democrático y este proyecto se alinea a las bases democráticas manteniendo presente la importancia de la transparencia electoral como un medio para promover la confianza en el proceso y fortalecer la democracia (*T-637-01 Corte Constitucional de Colombia, s/f*).

Ley 2108 del 29 de julio de 2021.

La Ley 2108 establece que el servicio de internet debe considerarse un servicio público esencial en Colombia. Esta legislación resalta la necesidad de brindar conectividad en todo el territorio nacional, especialmente en áreas rurales y comunidades en situación de vulnerabilidad. El proyecto se basa en esta ley para promover la inclusión digital y la transparencia en el proceso electoral (*Ley 2108 de 2021 - Gestor Normativo - Función Pública, s/f*).

Participación Ciudadana y Fortalecimiento Institucional.

Al proteger la decisión de los votantes y mitigar los riesgos de fraude electoral, se contribuye al fortalecimiento de la democracia y al fomento de una sociedad democrática sólida.

Así como el incremento de participación ciudadana en los procesos electorales y el fortalecimiento de las instituciones gubernamentales (*Combatir el fraude electoral ¿responsabilidad de todos o del Estado?* – Red Uva, s/f).

Riesgo de Fraude Electoral.

El riesgo de fraude electoral se refiere a la amenaza de manipulación de los resultados electorales. El análisis detallado realizado por la Misión de Observación Electoral (MOE) ha identificado áreas críticas en Colombia con un alto riesgo de fraude, lo que resalta la urgente necesidad de abordar estas preocupaciones (*Combatir el fraude electoral ¿responsabilidad de todos o del Estado?* – Red Uva, s/f).

Tecnologías de Desarrollo

JavaScript: Es un lenguaje de programación ampliamente utilizado en el desarrollo web. Se ejecuta en el navegador del usuario y se utiliza para crear interacciones dinámicas en las páginas web (Maza, 2012).

React.js: Es una biblioteca de JavaScript utilizada para la creación de interfaces de usuario. Permite el desarrollo de componentes reutilizables y facilita la construcción de aplicaciones de una sola página (SPA) eficientes (Diaz et al., 2017).

JSON (JavaScript Object Notation): Es un formato de intercambio de datos ligero y legible por humanos. Se utiliza para estructurar y transmitir datos entre el servidor y el cliente de manera eficiente (Aguirre, 2020).

Node.js: Es un entorno de ejecución de JavaScript del lado del servidor. Permite a los desarrolladores utilizar JavaScript tanto en el cliente como en el servidor, lo que simplifica el desarrollo de aplicaciones web complejas (Haro et al., s/f).

Express: Es un marco de desarrollo web para Node.js que simplifica la creación de aplicaciones web y la gestión de rutas y solicitudes HTTP. En otras palabras, Express es una biblioteca de Node.js que simplifica la construcción de aplicaciones web, proporcionando una estructura y funciones útiles para manejar solicitudes HTTP, enrutamiento y otras tareas comunes en el desarrollo web (Haro et al., 2019).

PostgreSQL: Es un sistema de gestión de bases de datos relacional que se destaca por su rendimiento y robustez. Se utiliza para almacenar y gestionar los datos de manera eficiente y segura (Ginestà et al., s/f).

Serverless: Paradigma de computación en el cual el desarrollador puede construir y ejecutar aplicaciones y servicios sin tener que gestionar la infraestructura. La gestión del servidor, capacidad de cómputo, y escalabilidad son manejadas automáticamente por el proveedor de la nube, lo que permite concentrarse en la solución de software y delegar el manejo a una plataforma como Vercel (*Qué es Serverless, ventajas y servicios / OpenWebinars, s/f*).

Interfaz de Programación de Aplicaciones - API: Es una interfaz uniforme, que permite que sistemas desarrollados con tecnologías diferentes, o diferentes arquitecturas se comuniquen, permitiendo una mayor versatilidad. Esta interfaz ha permitido la separación más eficiente del backend y el frontend, y promovido la popularidad de microservicios (*¿Qué es una API? - Explicación de interfaz de programación de aplicaciones - AWS, s/f*).

RESTful (Transferencia de Estado Representacional): Es un estilo de arquitectura de software para sistemas hipermedia distribuidos, utilizando métodos HTTP de forma semántica (como GET, POST, PUT, DELETE) para operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre recursos identificados a través de URLs (*¿Qué es una API de RESTful? - Explicación de API de RESTful - AWS, s/f*).

Vercel: Es una de las plataformas web más populares en el desarrollo de React, aunque no se limita solamente a esa librería. También Vercel es la empresa detrás de NextJS, uno de los frameworks de React más utilizados. Pero su servicios son más completos, ya que es en sí una plataforma de despliegue y alojamiento que permite a los desarrolladores publicar sitios web y aplicaciones con facilidad. Vercel se especializa en alojamiento para proyectos de Frontend y Serverless, ofreciendo escalabilidad automática, integración con Git y optimizaciones de rendimiento (*Vercel: Build and deploy the best Web experiences with The Frontend Cloud – Vercel, s/f*).

Cloudinary: Es una plataforma de gestión de activos multimedia en la nube que permite a los desarrolladores subir, almacenar, gestionar, manipular y entregar imágenes y videos en sitios web y aplicaciones. Cloudinary ofrece herramientas automáticas de optimización y transformación de estos activos para mejorar la carga y el rendimiento en diferentes dispositivos (*Image and Video Upload, Storage, Optimization and CDN, s/f*).

Metodología de prototipo

La metodología de prototipo es un enfoque de desarrollo que se basa en la creación iterativa y gradual de versiones tempranas y simplificadas de un sistema o producto. En el contexto de este proyecto, la metodología de prototipo se ha elegido para permitir una adaptación ágil a los requisitos cambiantes y a las condiciones del entorno de desarrollo. Esto significa que el equipo de desarrollo crea versiones iniciales del sistema de información en línea de manera incremental, lo que permite ajustes y mejoras continuas a medida que se obtienen comentarios de los usuarios y se identifican nuevas necesidades (Desarrollo De Software & Gabriel Pacienza, s/f).

Requisitos del Sistema

Recopilar y documentar los requisitos funcionales y no funcionales del sistema, incluidos los aspectos de seguridad y transparencia.

Generar informes con la información ingresada, para tener actualizada la cantidad de votos suministrada por candidato

Permitir filtrar la información agregada

Diseño del Prototipo

Elaborar un diseño de la arquitectura del sistema, dividiéndolo en componentes de backend y frontend.

Utilizar herramientas como Figma o Adobe XD para crear prototipos de la interfaz de usuario, enfocándose en la usabilidad y la experiencia del usuario.

Definir los endpoints de la API para la comunicación entre el frontend y el backend.

Desarrollo del Prototipo

Implementar el backend del sistema utilizando Node.js y Express, asegurando la integración con PostgreSQL para el almacenamiento de datos.

Desarrollar el frontend utilizando React y Tailwind CSS, siguiendo el diseño del prototipo de interfaz de usuario.

Implementar la lógica de negocio para recopilar y procesar los datos del preconteo electoral, garantizando la integridad y seguridad de la información.

Pruebas del Prototipo

Realizar pruebas unitarias para cada componente del sistema, utilizando herramientas como Jest para el backend y React Testing Library para el frontend.

Ejecutar pruebas de integración para verificar la comunicación adecuada entre el frontend y el backend.

Realizar pruebas de seguridad para identificar posibles vulnerabilidades y garantizar la protección de los datos del preconteo electoral.

Evaluación del Prototipo

Realizar ajustes y mejoras en función de la retroalimentación recibida, priorizando los aspectos críticos identificados durante la evaluación.

Preparar una presentación del prototipo y sus funcionalidades para su revisión y aprobación por parte de los interesados.

Recursos para el desarrollo del proyecto.

Software: Se requiere un IDE compatible con el desarrollo web, como Visual Studio Code o Atom.

Lenguajes de Programación: JavaScript para el desarrollo tanto del frontend como del backend. HTML y CSS para la estructura y el diseño de la interfaz de usuario.

Frameworks y Bibliotecas: React.js para el desarrollo del frontend, Node.js y Express.js para el desarrollo del backend, Tailwind CSS para el diseño y estilizado de la interfaz de usuario.

Gestión de Base de Datos: PostgreSQL como sistema de gestión de base de datos relacional.

Herramientas de Versionado: Git para el control de versiones del código fuente, con un repositorio en Github.

Herramientas de Pruebas: Postman.

Hardware:

- Disco Sólido de 256 GB.
- Procesador Intel Core i5 de 5 generación
- Memoria RAM de 8 GB

- Windows 10 +, MacOS 11 +, Ubuntu 22 latest + o distribución linux con kernel actualizado.

Requerimientos para la ejecución del proyecto

Servidor Web: Se requiere un servidor web compatible con Node.js y Express.js para ejecutar el backend del sistema.

Base de Datos: PostgreSQL debe estar instalado y configurado en el servidor para alojar la base de datos del sistema.

Navegador Web: Los usuarios necesitarán navegadores web modernos compatibles con HTML, CSS y JavaScript para acceder al frontend del sistema. Aunque el proyecto usa babel para garantizar que el código pueda funcionar en navegadores más antiguos.

Recurso humano para el desarrollo del proyecto:

Estudiante Desarrollador Full Stack: Responsable de desarrollar tanto el frontend como el backend del proyecto. Conocimientos básicos en HTML, CSS, JavaScript, React.js, Node.js y Express.js. Capacidad para aprender y aplicar nuevas tecnologías según sea necesario.

Tutor o Asesor Académico: Un profesor o tutor académico que pueda supervisar y orientar al estudiante durante el desarrollo de la tesis. Debe tener experiencia en el área de

sistemas de información y desarrollo de software. Proporcionará orientación en la planificación, diseño y ejecución del proyecto.

Recurso humano para la ejecución del proyecto.

La implementación de la solución tecnológica requerirá el siguiente personal para su óptimo funcionamiento.

Administrador del Sistema (Estudiante): Responsable de la gestión y mantenimiento del servidor donde se aloja la aplicación web. Debe tener conocimientos básicos de administración de sistemas operativos y servidores. Encargado de configurar y mantener actualizados los servicios necesarios para la ejecución del sistema, como Node.js, PostgreSQL, entre otros.

Desarrollador (Estudiante): Encargado de realizar pequeñas modificaciones o correcciones de errores en el sistema durante su ejecución.

Recursos económicos

A continuación se hace un cuadro informativo en donde se especifican los recursos económicos utilizados en el desarrollo del proyecto.

Tabla 1

Recursos económicos.

1. Recursos económicos estimados		
RECURSO	DESCRIPCIÓN	PRESUPUESTO
1. Equipo Humano	1 desarrollador Full Stack. Pago 2.500.000 mes.	12.500.000
2. Equipos y Software	1 computador portátil funcional. Internet x 5 meses	4.000.000
3. Viajes y Salidas de Campo	Ninguno	0
4. Académico	1 curso de desarrollo con React JS 1 curso de desarrollo con Express.js 1 curso de control de versiones con GitHub	300.000

5. Bibliografía

OWIUS. Elaboración de un presupuesto para desarrollo de software. 2022. <https://owius.com/elaboracion-de-un-presupuesto-de-desarrollo-de-software/>

Oficina de proyectos de informática. Ejemplo de presupuesto de un proyecto de software. 2022. <http://www.pmoinformatica.com/2018/05/ejemplo-presupuesto-proyecto-software.html>

TOTAL: 16.800.000

Diseño de la investigación

Entrevista: Entrevista para conocer las necesidades o los requerimientos del sistema.

Lugar: Yopal, Casanare.

Nombre del entrevistado: Licenciado Álvaro Reyes.

Fecha de la entrevista: 06/05/2023.

Preguntas que se realizaron:

Antecedentes y necesidades: ¿Cuáles son las principales necesidades o problemas que busca resolver con este sistema de seguimiento electoral?

Respuesta: Queremos cuidar la votación de nuestro candidato. Velar porque no se pierda ninguno de los votos en cada una de las mesas.

¿Qué experiencias previas o incidentes han motivado la búsqueda de una solución tecnológica para el seguimiento de resultados electorales en el departamento de Casanare?

Respuesta: Anteriormente pudimos implementar un centro de cómputo para las últimas elecciones atípicas para la alcaldía del municipio de Yopal, Casanare. En aquella oportunidad pudimos obtener la votación de cada uno de los candidatos y conocimos el ganador que fue por mucha diferencia. En aquella oportunidad ganó el candidato Leonardo Puentes y efectivamente la Registraduría publicó los mismos resultados. Esta vez queremos lo mismo pero a nivel del departamento para las elecciones a la gobernación de Casanare.

Funcionalidades requeridas: ¿Qué funcionalidades específicas espera que tenga el sistema de seguimiento de resultados electorales?

Respuesta: Queremos que a través de un celular se pueda cargar la información de los votos y también desde el celular ver los resultados. Así como realizar consultas y exportar la información.

¿Qué información o datos considera crucial que el sistema sea capaz de proporcionar en tiempo real?

Respuesta: Lo principal es que muestre la información de los votos de cada candidato y muestre él que va ganando. También que permita consultar los votos de los candidatos en cada mesa y exportar esa información en un PDF para hacer algún reclamo en la registraduría por si es el caso.

Interfaz y accesibilidad: ¿Qué requisitos tiene en cuanto a la interfaz de usuario del sistema?

Respuesta: Que la parte visual sea agradable y que no se cansen las vistas. Que sea fácil de manejar y muy básica para lo que se requiere.

¿Cómo le gustaría que los usuarios accedan y utilicen el sistema?

Respuesta: Que cada usuario sea responsable de la información que sube y la información que se elimine. Además el candidato es el único que puede ver los resultados y exportarlos.

Análisis de los requerimientos del sistema

Comprender correctamente los requisitos del sistema nos permite diseñar adecuadamente el sistema de información. A continuación, se desglosan los requisitos identificados en la entrevista.

a. Registro de Usuarios

- Descripción: El sistema debe permitir registrar usuarios.
- Consideraciones: Debe haber un flujo de registro que incluya la captura de información esencial, como nombre, correo electrónico y contraseña.

b. Registro de Resultados Electorales

- Descripción: El sistema deberá permitir el registro de votación en cada mesa.
- Consideraciones: Debe existir una interfaz que permita a los usuarios ingresar los resultados de manera precisa. Deben implementarse controles para asegurarse de que los datos sean coherentes y correctos al momento de enviarse.

c. Obtención de Resultados Electorales

- Descripción: El sistema debe mostrar la votación más alta entre los candidatos.
- El sistema debe permitir obtener los resultados electorales de acuerdo con los datos insertados en todas las mesas de votación.
- Consideraciones: Debe haber una función que calcule y presente los resultados de manera clara y comprensible. Los resultados deben ser actualizados y precisos.

d. Asignación de puestos de votación

- Descripción: Los usuarios encargados de registrar la información en el sistema solo tendrán asignado un puesto de votación.
- A cada usuario de tipo editor al momento de registrarse se le asignará un puesto de votación en el cual podrá registrar únicamente la información de las mesas contenidas en dicho puesto.
- Consideraciones: Una vez asignado el puesto de votación a cada usuario editor, no podrá tener acceso a ningún otro puesto de votación.

e. Roles de Administración.

- Descripción: Aunque no es una solicitud de requerimiento por parte del interesado, se debe realizar esta configuración de tal manera que cada rol tiene unos permisos establecidos.
- Consideraciones: Debe haber diferentes niveles de acceso y permisos para los usuarios con roles de administración. Por ejemplo, un administrador principal podría tener más privilegios que un administrador de nivel inferior, un usuario "Editor" sólo puede tener acceso a registrar información, un usuario de tipo "candidato" solo puede tener acceso a consultar la información registrada en las mesas y observar los resultados de la votación en general.

f. Consultas rol candidato:

- Descripción: Solo los candidatos podrán consultar la información del sistema.
- El sistema debe contener toda la información de las mesas registradas y debe permitirles únicamente a los usuarios candidatos consultar la información registrada en las mesas.

- Consideraciones: Se le debe asignar el rol “Candidato” a estos usuarios.

g. Consultas por municipio, mesas y puestos de votación:

- El sistema debe permitir realizar consultas por municipios, mesas y puestos de votación y mostrar la votación obtenida por cada candidato, votos en blanco, votos nulos o votos no marcados según la consulta.
- Consideraciones: Se debe proporcionar una interfaz que permita a los usuarios seleccionar un municipio, una mesa o un puesto de votación para obtener los resultados asociados a la consulta. Los datos deben ser filtrados y presentados de manera legible.

h. Evidencia del formato E14:

- El sistema debe permitir guardar la imagen del formato E14 generada por los jurados de votación (*Jurados de votación: Cómo llenar formulario E-14 en elecciones presidenciales / Elecciones 2022 / Portafolio, s/f*).
- Consideraciones: Debe existir un campo de imagen en el formulario de envío de datos para cada mesa de votación.

i. Exportar la información:

- El sistema debe permitir exportar la información en formato Excel de acuerdo con la consulta que se realice.
- Consideraciones: Se debe crear un módulo que permita exportar todos los campos junto con su información de una forma organizada.

j. Accesibilidad:

- El sistema debe ser accesible para cualquier resolución de pantalla (*¿Qué es la accesibilidad? - Aprende desarrollo web / MDN, s/f*).

- Consideraciones: El diseño y la interfaz del sistema deben ser responsivos, adaptándose automáticamente a diferentes tamaños de pantalla y dispositivos.

k. Seguridad de la Información:

- Descripción: El sistema debe contar con cierto nivel de seguridad para proteger la información registrada en la base de datos.
- Consideraciones: Deben implementarse medidas de seguridad, como autenticación de usuarios, encriptación de datos y controles de acceso, para garantizar que la información esté protegida contra accesos no autorizados (*¿Qué es la seguridad de aplicaciones? / Akamai, s/f*).

Encuesta en línea para la viabilidad del proyecto.

La encuesta de opinión pública se llevó a cabo con el fin de evaluar la viabilidad y el impacto potencial de la propuesta de diseñar un sistema de información mediante la metodología de prototipo para el seguimiento de los resultados de preconteo electoral de las elecciones regionales para la gobernación del departamento de Casanare, especialmente en respuesta a las preocupaciones sobre el riesgo de fraude electoral y la necesidad de fortalecer la transparencia y garantizar la decisión de los ciudadanos votantes.

La encuesta se diseñó con cuatro preguntas cerradas y una pregunta abierta teniendo como base el tema del riesgo electoral, la credibilidad de los ciudadanos en las instituciones encargadas de divulgar esta información y la posibilidad de incluir nuevas tecnologías en los procesos electorales en el departamento.

Título de la encuesta: Encuesta de opinión pública para la viabilidad de implementar un sistema de seguimiento a los resultados del preconteo electoral, en las elecciones a la gobernación en el departamento de Casanare.

Enlace de la encuesta: <https://forms.gle/6p7GyVdRPNti5RD38>

Resultados de la encuesta:

https://docs.google.com/forms/d/1jSEllrKUDm1uoCOaM1S0GJx2I_0xgd4hSaHoYdgzwoY/viewanalytics

Análisis de datos de la encuesta: A continuación se analizan cada una de las respuestas a las preguntas presentes en la encuesta.

Percepción de la confiabilidad de los resultados electorales: La mayoría de los encuestados considera que los resultados electorales entregados por la Registraduría Nacional para las elecciones en el departamento de Casanare son confiables en cierta medida. Solo un pequeño porcentaje los considera muy confiables. Lo que da a entender que más del 50% de los ciudadanos encuestados no creen totalmente en los resultados electorales que se publican.

Opinión sobre el seguimiento de los resultados de preconteo electoral: Hay un consenso entre los encuestados de que realizar un seguimiento en primera instancia de los resultados de preconteo electoral en las elecciones de Casanare puede contribuir a disminuir el riesgo de fraude electoral. La mayoría está de acuerdo con esta afirmación.

Importancia de incluir nuevas tecnologías en los procesos electorales: La gran mayoría de los encuestados está de acuerdo en que incluir nuevas tecnologías en los procesos electorales para realizar un seguimiento a los resultados publicados por los jurados de votación en cada mesa instalada en el departamento de Casanare es importante para garantizar y respetar la decisión de los ciudadanos en las urnas.

Medidas adicionales para garantizar la transparencia y garantizar la decisión de los ciudadanos votantes: Respecto a medidas adicionales que podrían implementarse para

garantizar la transparencia en los procesos electorales del departamento, se sugiere la instalación de mesas de votación de metal y completamente selladas después de que los jurados publiquen los resultados de preconteo, así como el uso de la biometría como medida adicional tecnológica.

Conclusión de la encuesta: Los resultados de la encuesta según el análisis a cada pregunta arrojaron que existe cierta preocupación por la confiabilidad de los resultados electorales y una fuerte creencia en la importancia de utilizar nuevas tecnologías para garantizar la transparencia y garantizar la decisión de los ciudadanos votantes. Lo cual nos indica que existe la posibilidad de que un proyecto de esta característica pueda ser aplicado o llevado a cabo.

Diseño gráfico para interpretar los requisitos del sistema.

En base a los requisitos iniciales se creó un diagrama de flujo para identificar las características esenciales del sistema y posteriormente diseñar un modelo entidad relación (MER) para definir las tablas o entidades a utilizar en la base de datos y además definir las posibles relaciones entre sí. (alg/bd & 2012, s/f).

Figura 4

Diagrama de flujo para identificar las posibles entidades en la base de datos

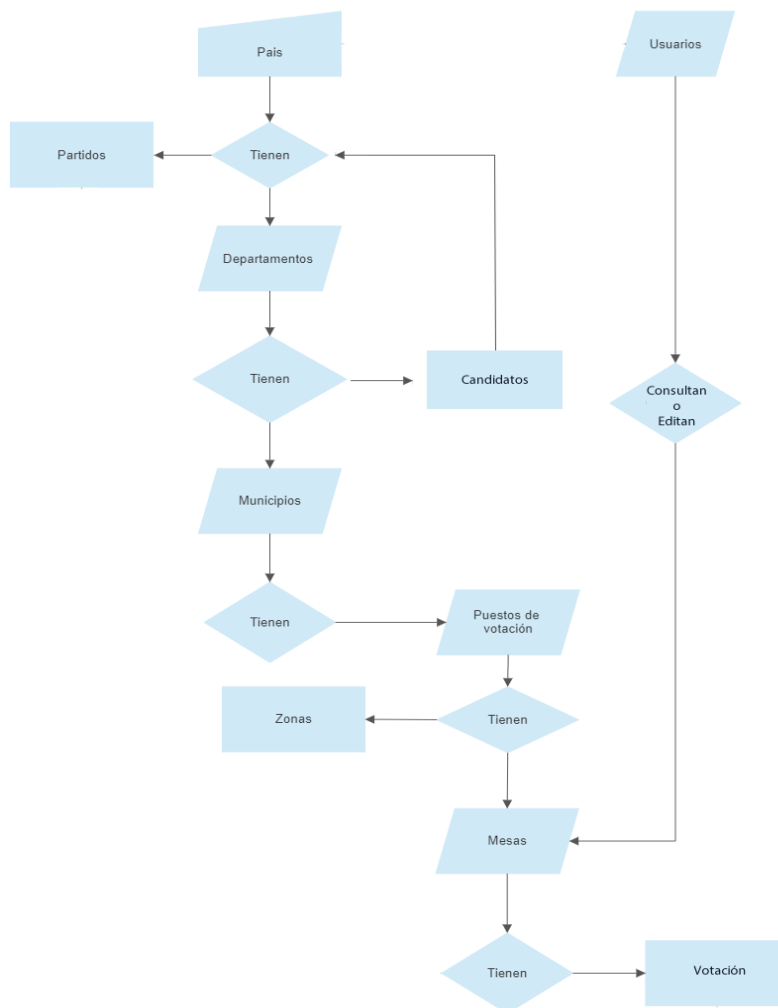
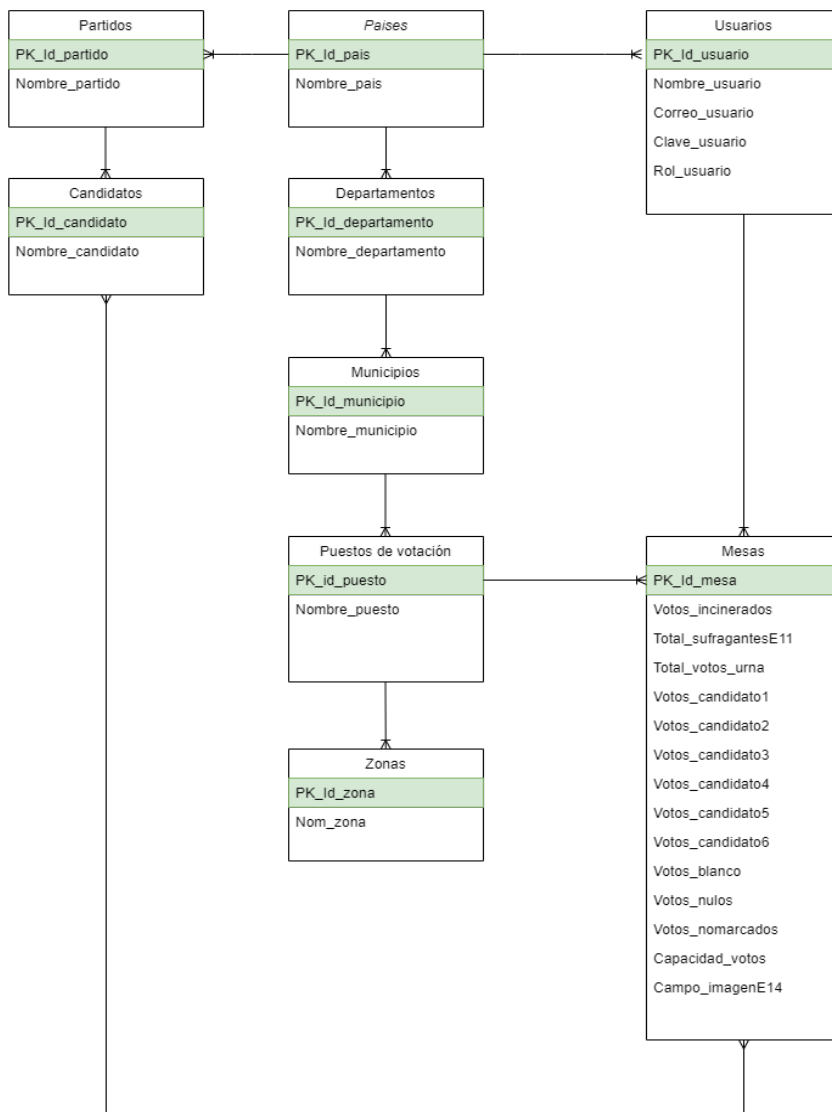


Diagrama de clases:

El diagrama de clase se construyó teniendo como referencia el diagrama de flujo anterior y se añadieron algunos de los campos correspondientes para cada una de las clases.

Figura 5

Descripción del diagrama de clases



Descripción de las clases:

A continuación, se describe cada una de las clases (tablas) teniendo como referencia el diagrama de clases.

Las clases definidas en esta instancia son las tablas que se crearán en la base de datos.

Tabla 2

Descripción de la clase usuarios.

Clase	Usuarios
Descripción	Esta clase permite a los usuarios editar su información personal, así como consultar o registrar la información del E14 en cada una de las mesas, de acuerdo con su rol (administrador, editor, candidato).
Operaciones	<p>Consultar: Los usuarios con el rol de candidato podrán ingresar al sistema y observar los resultados electorales desde su perfil, así como realizar consultas.</p> <p>Registrar: Los usuarios con el rol de editor podrán ingresar al sistema y registrar la información de cada E14.</p> <p>Administrar: Los usuarios con el rol de administrador podrán consultar, registrar y modificar información en el sistema. Tendrán acceso total al sistema.</p>

Nota: En la tabla 2 se evidencia la descripción y operaciones de la clase usuarios. Fuente. Autor

Tabla 3

Descripción de la clase mesas.

Clase	Mesas
Descripción	La función de la clase mesas es recolectar la información de cada E14. Previamente debe estar creada en la base de datos.
Operaciones	Guardar: La única operación de esta clase es guardar la información que le envían los usuarios editores.

Tabla 4

Descripción de la clase puestos de votación.

Clase	Puestos de votación
Descripción	Esta clase debe contener la información de todos los puestos de votación habilitados para las elecciones.
Operaciones	Guardar: La única operación de esta clase es guardar la información que se genera desde el script de la base de datos; donde están escritos todos los puestos de votación.

Tabla 5

Descripción de la clase candidatos.

Clase	Candidatos
Descripción	Esta clase contiene todos los nombres de los candidatos inscritos en el tarjetón para la gobernación de Casanare.

Operaciones	Guardar: La única operación de esta clase es guardar la información que se genera desde el script de la base de datos; donde están escritos todos los candidatos inscritos en el tarjetón.
--------------------	---

Tabla 6

Descripción de la clase partidos.

Clase	Partidos
Descripción	Esta clase contiene todos los nombres de los partidos políticos inscritos en el tarjetón para la gobernación de Casanare.
Operaciones	Guardar: La única operación de esta clase es guardar la información que se genera desde el script de la base de datos; donde están escritos todos los partidos políticos inscritos en el tarjetón de la gobernación de Casanare.

Tabla 7

Descripción de la clase municipios.

Clase	Municipios
Descripción	Esta clase contiene todos los nombres de los municipios del departamento de Casanare.
Operaciones	Guardar: La única operación de esta clase es guardar la información que se genera desde el script de la base de datos; donde están escritos todos los municipios del departamento de Casanare.

Tabla 8

Descripción de la clase zonas.

Clase	Zonas
Descripción	Esta clase contiene todos los nombres de las zonas (rural, urbana) para identificar la zona en que se encuentra cada puesto de votación.
Operaciones	Guardar: La única operación de esta clase es guardar la información que se genera desde el script de la base de datos; donde están escritas las zonas urbanas o rurales.

Normalización de la base de datos.

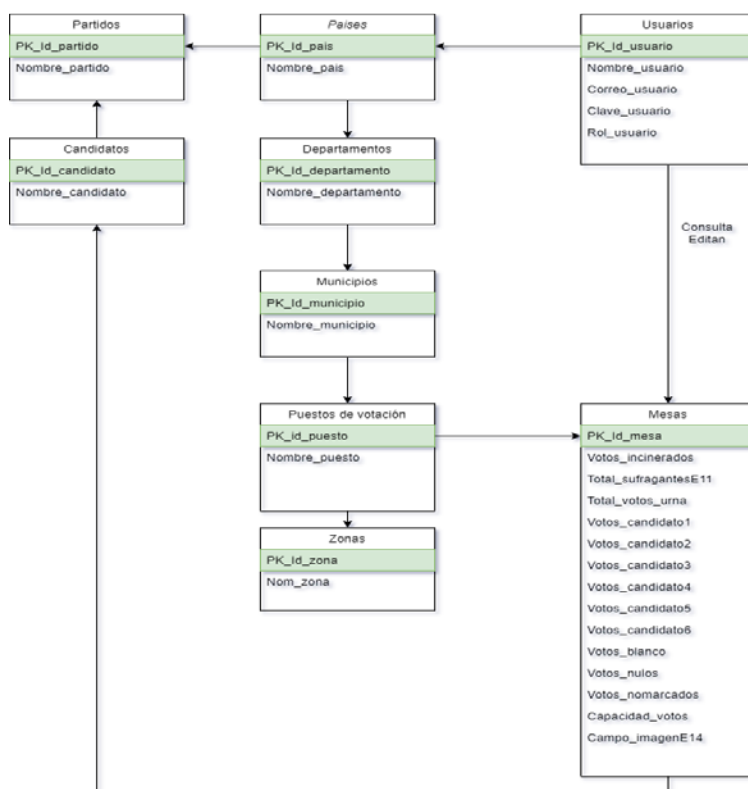
La normalización de la base de datos se hizo con el fin de asignarle a cada una de las tablas presentes en la base de datos un identificador único, relacionar tablas entre sí y eliminar la redundancia de datos. Es decir, que no se repita información en las tablas porque todo esto genera mayor tamaño de la base de datos; ya sea en la nube o localmente (Arias, 2014).

Primera forma normal:

Esta primera forma normal se llevó a cabo para seleccionar de cada tabla un identificador único.

Figura 6

Normalización de base de datos. primera forma normal



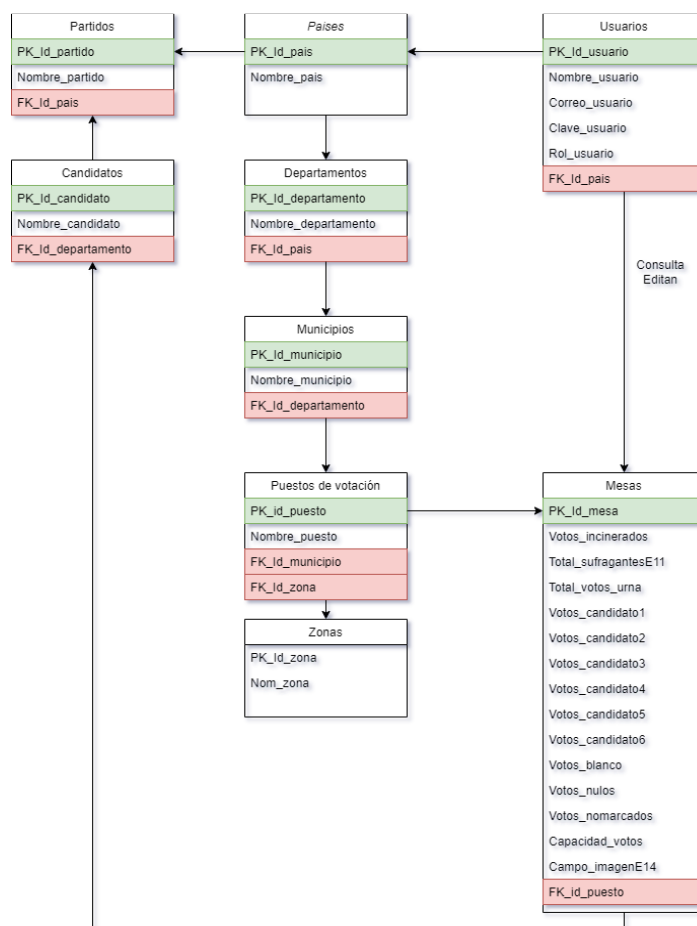
Segunda y tercera forma normal:

El proceso de la segunda y tercera forma normal se realizó para identificar los campos foráneos en tablas relacionadas entre sí y eliminar duplicados de información. Generalmente una base de datos está normalizada en su tercera forma normal.

Esto permite tener una integridad de datos y permitir la coherencia de información en todo el sistema (Valencia Cabrera, s/f).

Figura 7

Normalización de base de datos. Segunda y tercera forma normal



Diagramas de casos de uso:

Los diagramas de casos de uso se diseñaron para interpretar cómo interactúan los usuarios con el sistema. Es decir, es el diseño del paso a paso que hace un usuario para enviar o consultar información en el sistema (Gutiérrez Rodríguez, s/f).

Caso de uso actores: Los diagramas de casos de uso de actores se crearon para identificar las personas que van a ejecutar el sistema y cómo interactúan con él de acuerdo con su rol.

A continuación, se representan los actores involucrados de acuerdo con su rol, los cuales son los encargados de registrar y consultar la información en el sistema.

Figura 8

Diagrama de caso de uso actores. El rol de administrador tendrá acceso a todas las funciones del sistema y su función principal es la de crear usuarios. El rol de editor solo registra usuarios y el rol de candidato solo puede consultar los datos ya agregados por el administrador y editor.



Diagrama de caso de uso general: Inicio de sesión.

Figura 9

Diagrama de caso de inicio de sesión



El usuario ingresa sus datos para iniciar sesión, el sistema realiza la validación de usuario y contraseña devolviendo un token de autenticación, si los datos son correctos el usuario podrá ingresar a la plataforma de administración.

Tabla 9

Descripción del caso de uso, inicio de sesión.

RF - 01 (Referencia)	Inicio de sesión
Objetivos asociados	El usuario debe estar registrado en la aplicación.
Requisitos asociados	Diligenciar formulario de inicio de sesión
Descripción	El usuario ingresa al enlace de la página web y diligencia los campos de inicio de sesión e ingresa al sistema.
Precondición	El usuario debe estar registrado en la base de datos
Secuencia Normal	Paso Acción

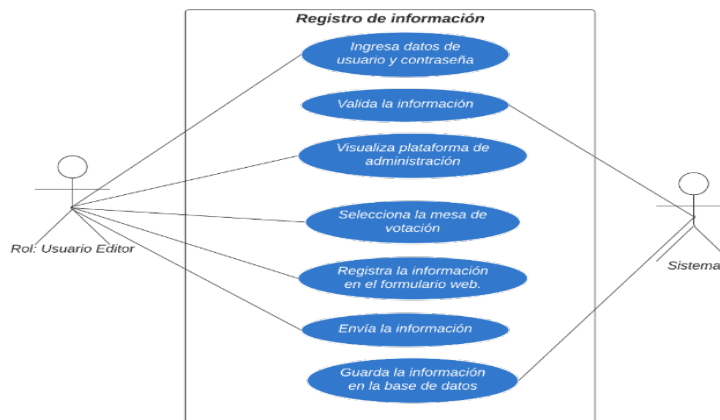
	1	El usuario ingresa a la página web.
	2	El usuario diligencia el formulario de inicio de sesión
	3	El usuario envía la información suministrada.
	4	El sistema valida la información enviada por el usuario.
	5	El sistema muestra el área de administración del sistema.
Postcondición		El usuario debe digitar la información correctamente.
Excepciones	Paso	Acción
	1	Ninguna.
	2	
Comentarios		Solo los usuarios registrados podrán ingresar al sistema.

La tabla indica el proceso que realiza el usuario para autenticarse en el sistema, ingresando sus datos y siendo validados por el backend, al ser los datos correctos se redirige al usuario a su dashboard o área de administración.

Diagrama de caso de uso: Registro de información.

Figura 10

Caso de uso registro de información



El diagrama presenta al rol editor autenticándose en el sistema e ingresando datos en una mesa de votación a través de un formulario. Estos datos al ser correctamente diligenciados se guardan en la base de datos.

Tabla 10

Descripción del caso de uso registro de información.

RF - 02 (Referencia)	Registro de información																				
Objetivos asociados	El usuario debe estar registrado en la aplicación.																				
Requisitos asociados	Diligenciar formulario de inicio de sesión																				
Descripción	El usuario ingresa al link de la página web y diligencia los campos de inicio de sesión e ingresa al sistema. Posteriormente selecciona la mesa de votación y digita la información del formato E14.																				
Precondición	El usuario debe estar registrado en la base de datos y tener el rol de editor.																				
Secuencia Normal	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El usuario ingresa a la página web.</td> </tr> <tr> <td>2</td> <td>El usuario diligencia el formulario de inicio de sesión</td> </tr> <tr> <td>3</td> <td>El usuario envía la información suministrada.</td> </tr> <tr> <td>4</td> <td>El sistema valida la información enviada por el usuario.</td> </tr> <tr> <td>5</td> <td>El sistema muestra el área de administración del sistema.</td> </tr> <tr> <td>6</td> <td>El usuario selecciona la mesa de votación</td> </tr> <tr> <td>7</td> <td>El sistema muestra el formulario del formato E14 con sus campos vacíos.</td> </tr> <tr> <td>8</td> <td>El usuario digita la información en el formulario E14.</td> </tr> <tr> <td>9</td> <td>El usuario envía la información diligenciada en el formulario E14.</td> </tr> </tbody> </table>	Paso	Acción	1	El usuario ingresa a la página web.	2	El usuario diligencia el formulario de inicio de sesión	3	El usuario envía la información suministrada.	4	El sistema valida la información enviada por el usuario.	5	El sistema muestra el área de administración del sistema.	6	El usuario selecciona la mesa de votación	7	El sistema muestra el formulario del formato E14 con sus campos vacíos.	8	El usuario digita la información en el formulario E14.	9	El usuario envía la información diligenciada en el formulario E14.
Paso	Acción																				
1	El usuario ingresa a la página web.																				
2	El usuario diligencia el formulario de inicio de sesión																				
3	El usuario envía la información suministrada.																				
4	El sistema valida la información enviada por el usuario.																				
5	El sistema muestra el área de administración del sistema.																				
6	El usuario selecciona la mesa de votación																				
7	El sistema muestra el formulario del formato E14 con sus campos vacíos.																				
8	El usuario digita la información en el formulario E14.																				
9	El usuario envía la información diligenciada en el formulario E14.																				

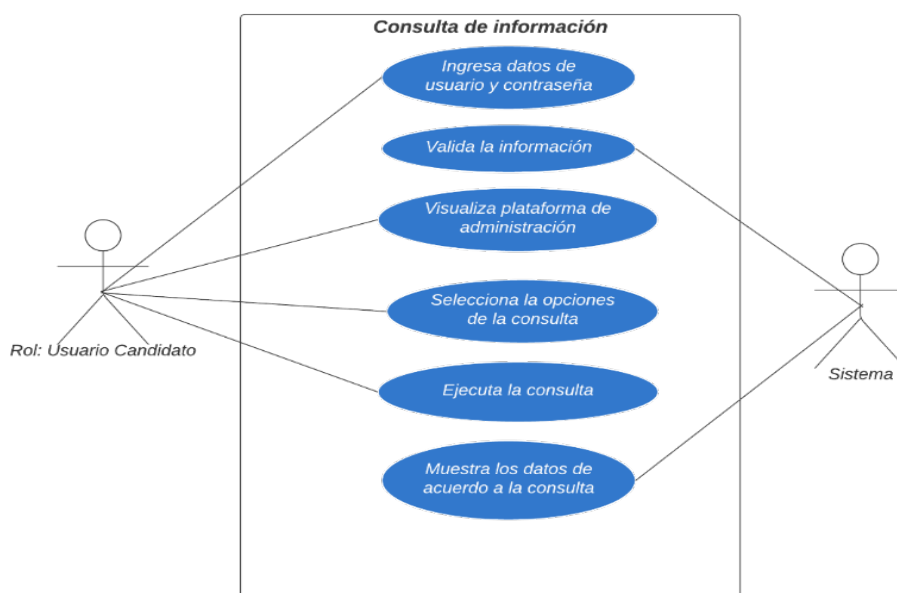
	10	El sistema valida el envío y guarda la información de la mesa en la base de datos.
Postcondición		El usuario debe verificar la información suministrada.
Excepciones	Paso	Acción
	1	Ninguna.
	2	
Comentarios		Solo los usuarios registrados con el rol de editor podrán registrar información en la base de datos.

La tabla presenta el acceso al sistema por parte del usuario editor, el cuál registra nuevos datos y estos al ser validados por el sistema se guardan en la base de datos.

Diagrama de caso de uso: Consultar información

Figura 11

Caso de uso, consulta de información



El rol candidato se autentica en el sistema, el cuál valida sus datos de inicio de sesión y lo redirige al dashboard, donde el usuario elige el municipio, puesto de votación y mesa a consultar. Los datos de la consulta son retornados desde la base de datos.

Tabla 11

Descripción del caso de uso consulta de información.

RF - 03 (Referencia)	Consulta de información	
Objetivos asociados	El usuario debe estar registrado en la aplicación.	
Requisitos asociados	Diligenciar formulario de inicio de sesión	
Descripción	El usuario ingresa al enlace de la página web y diligencia los campos de inicio de sesión e ingresa al sistema. Posteriormente selecciona la consulta (municipio, mesa, puesto de votación) y envía la consulta.	
Precondición	El usuario debe estar registrado en la base de datos y tener el rol de candidato.	
Secuencia Normal	Paso	Acción
	1	El usuario ingresa a la página web.
	2	El usuario diligencia el formulario de inicio de sesión
	3	El usuario envía la información suministrada.
	4	El sistema valida la información enviada por el usuario.
	5	El sistema muestra el área de administración del sistema.
	6	El usuario selecciona los filtros del formulario de búsqueda.
	7	El usuario envía la consulta.
	8	El sistema muestra la información de la consulta.
Postcondición	El usuario debe seleccionar los filtros del formulario de búsqueda.	
Excepciones	Paso	Acción
	1	El usuario no selecciona filtros de búsqueda.

2

Comentarios	Solo los usuarios registrados con el rol de candidato podrán consultar información en el sistema.
--------------------	---

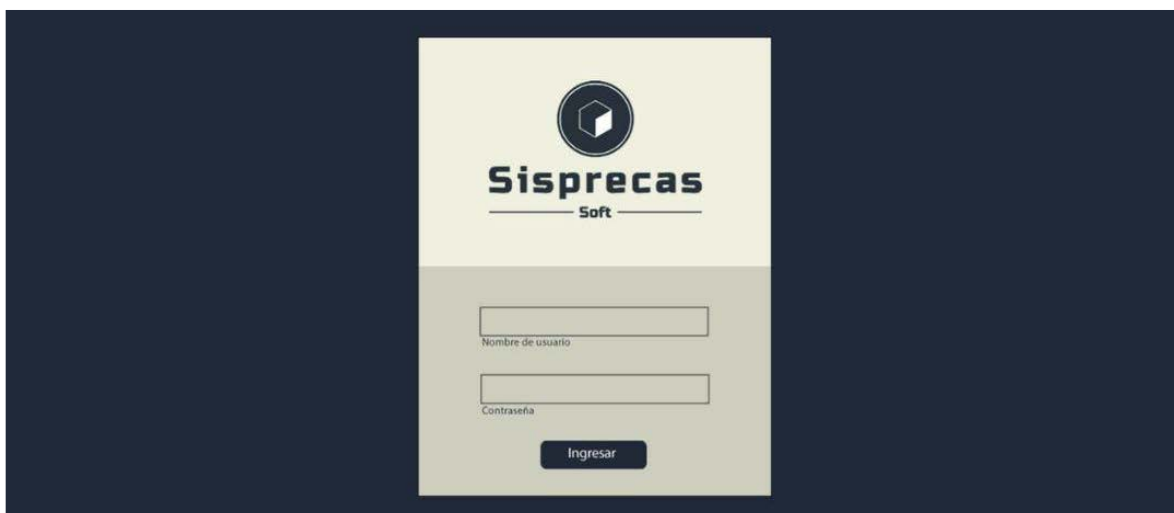
El usuario candidato se autentica en el sistema y verifica desde su dashboard el municipio, puesto de votación y/o mesa a consultar. Los datos son procesados desde la base de datos y presentados al usuario.

Diseño de la interfaz de usuario (UI).

El diseño se ha inspirado en el aplicativo que usa la Registraduría Nacional del Estado Civil Colombiano (*Software de Escrutinio, una herramienta tecnológica de la Registraduría Nacional al (...) - Registraduría Nacional del Estado Civil, s/f*)

Figura 12

Formulario de inicio de sesión



Fuente. Autoría propia

Figura 13

Panel de administración de usuarios editores



Panel de administración de usuarios editores. El panel muestra un encabezado con un ícono de casa, el título "Votación" y un perfil de usuario. A la izquierda, un recuadro muestra la información de ubicación: Departamento: Casanare, Municipio: Yopal, Puesto de votación: 1, Lugar: Colegio Luis Hernandez Vargas. A la derecha, una cuadrícula de 27 botones numerados del 01 al 27.

Departamento: Casanare	01	02	03	04	05	06	07	08	09
Municipio: Yopal	10	11	12	13	14	15	16	17	18
Puesto de votación: 1	19	20	21	22	23	24	25	26	27
Lugar: Colegio Luis Hernandez Vargas									

Fuente. Autoría propia

Figura 14

Formulario E-14

Registro de votación

Nota: Por favor verifique la información del formato E14 antes de enviarla. Recuerda que cada envío de información es responsabilidad de usted.

Departamento: Casanare Municipio: Yopal Puesto: 01 Mesa: 01

TOTAL SUFRAGANTES FORMATO E11	TOTAL VOTOS EN LA URNA	TOTAL VOTOS INCINERADOS
380	201	0

	Partido - Candidato	Votación
1	Cesar Augusto Ortiz Zorro	80
2	Luis Alexis García Barrera	10
3	Héctor Manuel Balaguera Q.	15
4	Jacobo Rivera Gómez	5
5	Alba Rocío Romero García	5
6	Joel Olmos Cordero	20
7	Rubiela Benitez Enriquez	5
8	Guillermo Alexander Velandia G.	10
9	Marisela Duarte Rodriguez	50

Votos en blanco	1
Votos Nulos	0
Votos no marcados	0

Imagen E14

Enviar

Fuente. Autoría propia

Figura 15

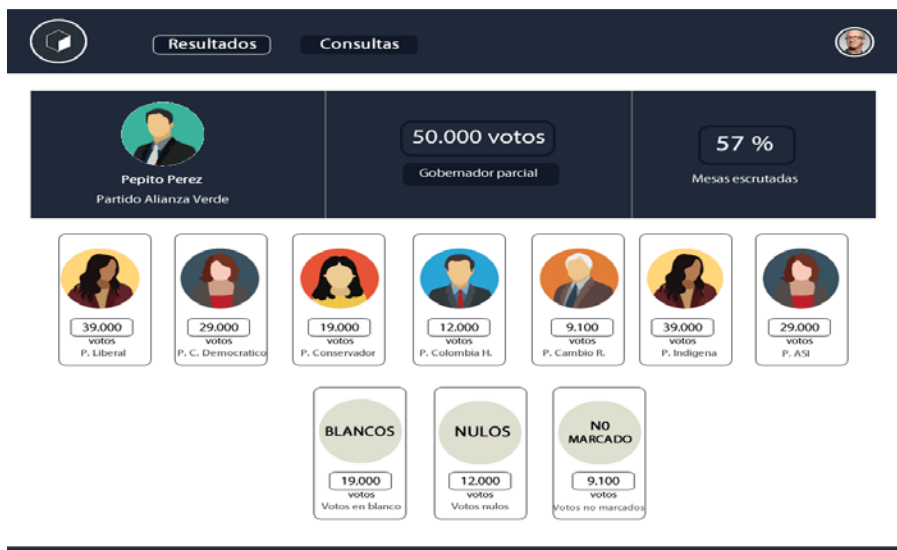
Panel de administración. Página de consultas



Fuente. Autoría propia

Figura 16

Panel de administración. Página de resultados



Fuente. Autoría propia

Desarrollo del prototipo

Actualmente existen muchas maneras de implementar software, con diferentes metodologías, infraestructuras, arquitecturas de software y patrones de diseño, lo cual permite personalizar el desarrollo a la necesidad y contexto propio del mismo.

SISPRECAS-Soft se ha desarrollado usando la metodología de prototipos, que permite presentar de manera más rápida resultados de desarrollo y con base en ellos ir avanzando hacia la aplicación solicitada. La inmediatez de este desarrollo implica que se optimicen los tiempos, por lo cual se usó la tecnología serverless, que nos permite concentrarnos más en el desarrollo del aplicativo y no en la infraestructura, delegando esta responsabilidad a Vercel, una empresa líder en el desarrollo e implementación de aplicaciones web modernas.

SISPRECAS-Soft también se ha diseñado dividiendo el proyecto entre backend y frontend, siendo la tecnología de backend Express JS, un framework que se ha consolidado en la industria y la tecnología del frontend, React JS, la librería que es una de las más importantes en el desarrollo web.

Debido al uso de Serverless en Vercel, se presentó la necesidad de tener una forma de almacenar eficientemente las imágenes de los formulario E14 que subiesen los usuarios encargados de enviar la información de cada mesa de votación a través de SISPRECAS-Soft, para lo cual se integró la solución frontend con Cloudinary, una plataforma líder en almacenamiento de recursos multimedia y se unificaron en el proyecto a través de la API que nos suministra esta plataforma, lo que permitió un manejo eficiente de las imágenes, ya que éstas se

suben a Cloudinary automáticamente y en la base de datos solo se almacena la url de cada imagen alojada en Cloudinary.

Para la comunicación entre el frontend y el backend, se ha diseñado una API con Express JS, que facilita el intercambio de datos entre el frontend y el backend, ya que en cada respuesta del backend a una solicitud de la aplicación de React JS, el primero devuelve un JSON que el segundo consume.

La persistencia de datos se realizó usando un motor de bases de datos de los más robustos y open source, como es PostgreSQL, el cual se implementa en Vercel y se comunica con el backend, para servir los datos almacenados de las elecciones regionales.

Además para la presentación de la información del lado del cliente, se integró TailwindCSS, que es un framework CSS, que permite el uso de CSS con clases de manera más eficiente, esto permitió disminuir los tiempos de diseño de SISPRECAS-Soft.

Estructura de la aplicación

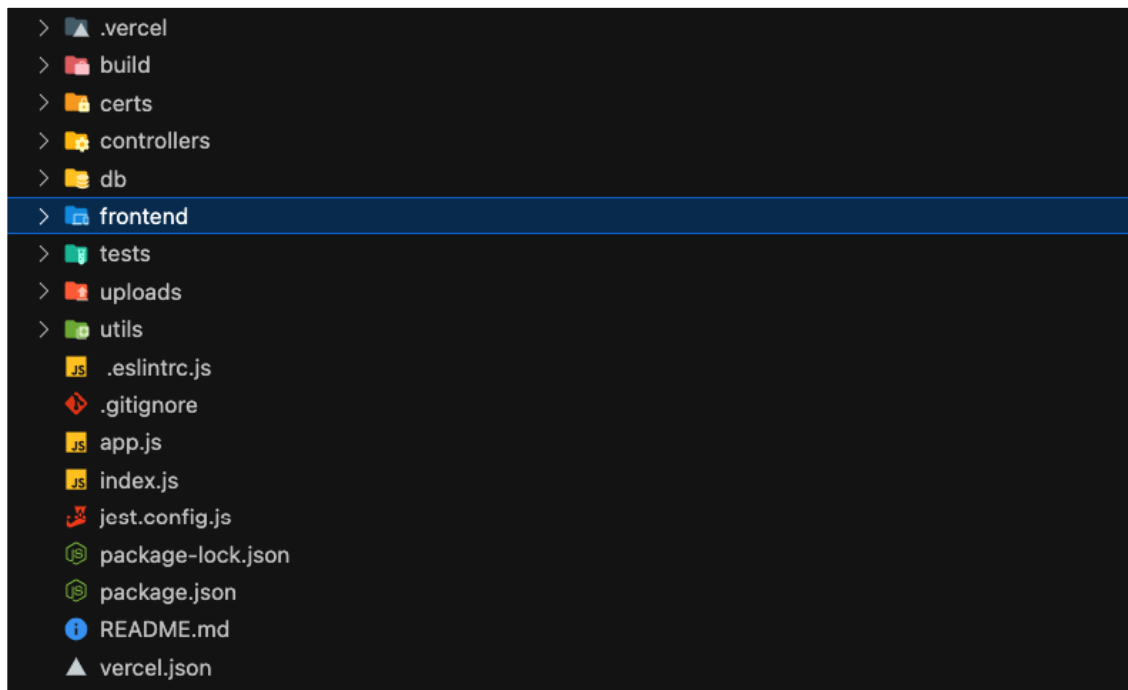
Backend: El backend del proyecto ha sido desarrollado con Express JS un framework de Node JS, que permite implementar un manejo de rutas eficiente, seguro, moderno y que implementa la arquitectura REST para la comunicación uniforme entre las diferentes partes de un aplicativo (Express - Infraestructura de aplicaciones web Node.js, s/f). El aplicativo de Express genera una comunicación con la capa de datos, desarrollada con PostgreSQL, un motor de base de datos robusto y open source, que usa el estándar SQL (Gibert et al., s/f)

FrontEnd

El frontend se desarrolló usando React JS, una de las tecnologías con mayor impacto actual en el desarrollo web, que permite un flujo más eficiente en el aplicativo, haciendo uso de tendencias y buenas prácticas (React, s/f).

Figura 17

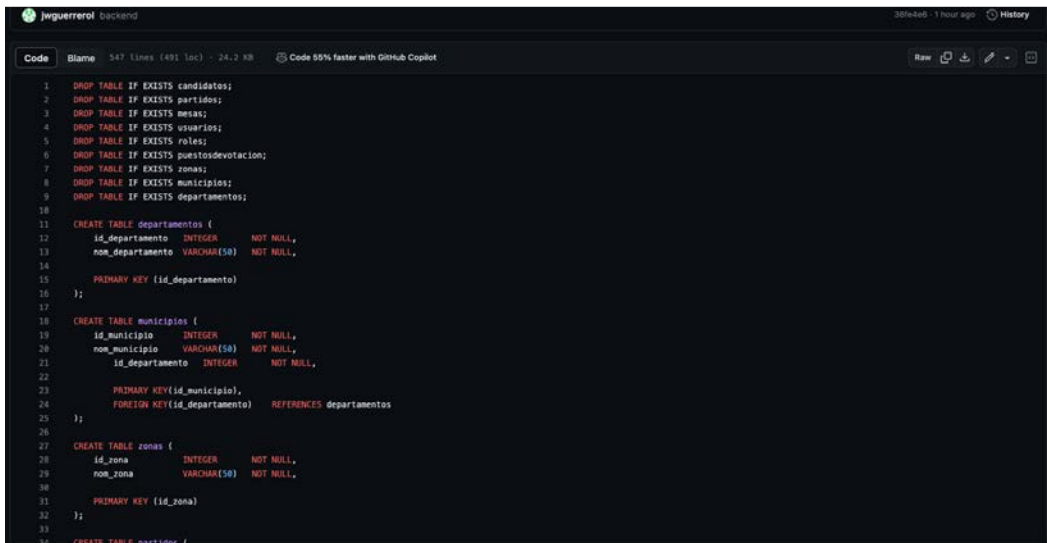
Estructura del proyecto



Fuente. Autoría propia

Figura 18

Script principal de la base de datos



```

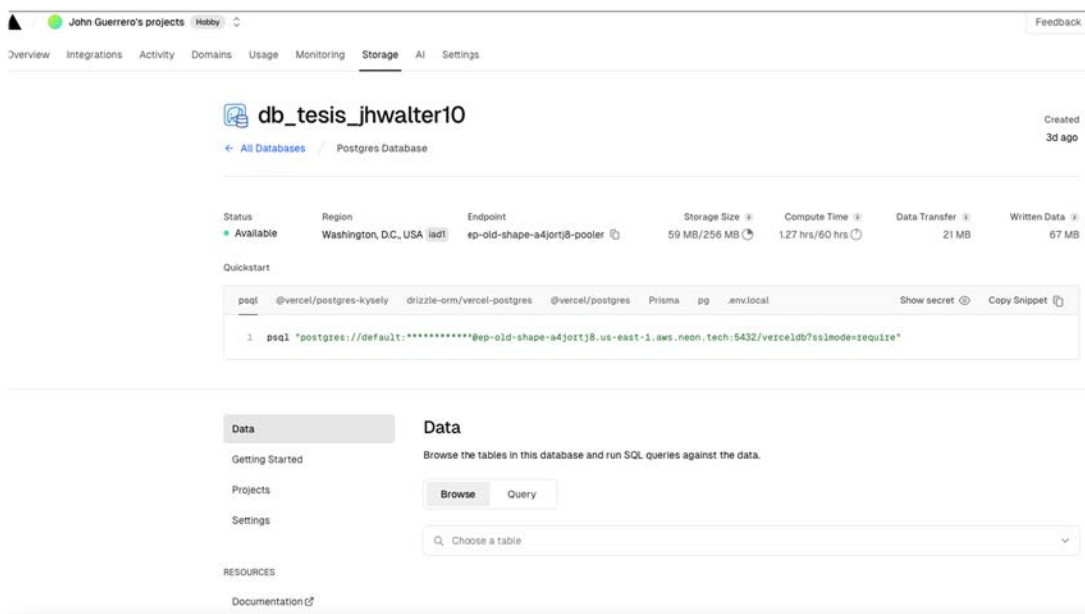
1 DROP TABLE IF EXISTS candidatos;
2 DROP TABLE IF EXISTS partidos;
3 DROP TABLE IF EXISTS mesas;
4 DROP TABLE IF EXISTS usuarios;
5 DROP TABLE IF EXISTS roles;
6 DROP TABLE IF EXISTS puestosdevotacion;
7 DROP TABLE IF EXISTS zonas;
8 DROP TABLE IF EXISTS municipios;
9 DROP TABLE IF EXISTS departamentos;
10
11 CREATE TABLE departamentos (
12   id_departamento INTEGER NOT NULL,
13   nom_departamento VARCHAR(50) NOT NULL,
14
15   PRIMARY KEY (id_departamento)
16 );
17
18 CREATE TABLE municipios (
19   id_municipio INTEGER NOT NULL,
20   nom_municipio VARCHAR(50) NOT NULL,
21   id_departamento INTEGER NOT NULL,
22
23   PRIMARY KEY (id_municipio),
24   FOREIGN KEY (id_departamento) REFERENCES departamentos
25 );
26
27 CREATE TABLE zonas (
28   id_zona INTEGER NOT NULL,
29   nom_zona VARCHAR(50) NOT NULL,
30
31   PRIMARY KEY (id_zona)
32 );
33
34 CREATE TABLE partidos (

```

Fuente. Autoría propia

Figura 19

Creación de la base de datos en Vercel



The screenshot shows the Vercel dashboard for a PostgreSQL database instance named `db_tesis_jhwalter10`. The instance is located in the Washington, D.C., USA region and is currently available. Key metrics include a storage size of 59 MB / 256 MB, a compute time of 1.27 hrs / 60 hrs, and a data transfer of 21 MB. The instance was created 3 days ago.

Below the instance details, there is a "Quickstart" section with a terminal snippet for connecting to the database:

```
psql @vercel/postgres-kyseiy drizzle-orm/vercel-postgres @vercel/postgres Prisma pg .env.local Show secret Copy Snippet
```

The terminal snippet shows the following command:

```
1 psql "postgres://default:*****@ep-ql-d-shape-a4jortj8.us-east-1.aws.neon.tech:5432/verceldb?sslmode=require"
```

At the bottom of the dashboard, there is a "Data" section with a "Browse" button and a search bar for selecting a table.

Fuente. Autoría propia

BackEnd

Figura 20

Ejecución del backend del proyecto backend en local



```
PROBLEMS OUTPUT TERMINAL PORTS

○ johnguerreroDESKTOP-83PRR0~/uni/sispreca npm run dev

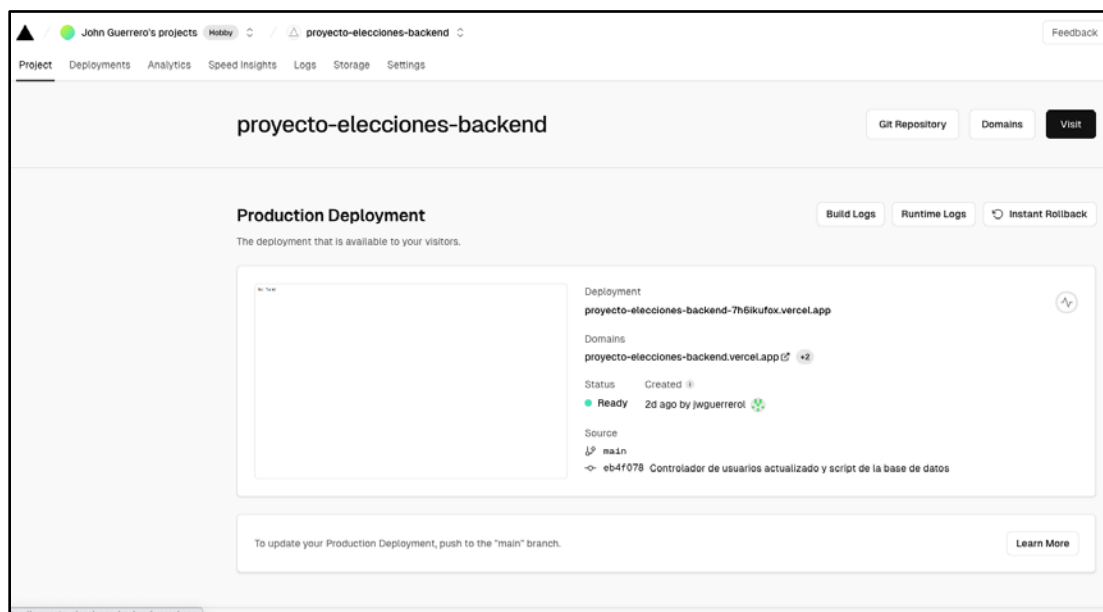
> jproject-backend@1.0.0 dev
> cross-env NODE_ENV=development nodemon index.js

[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
Server HTTP running on port 4000
connected to postgresql
```

Fuente. Autoría propia

Figura 21

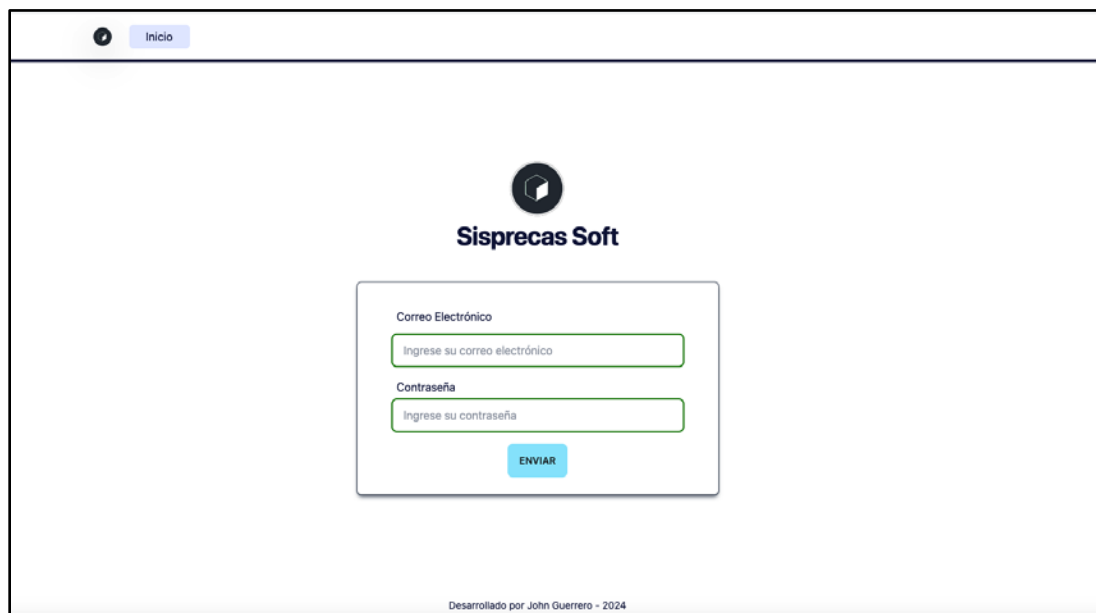
Proyecto backend ejecutado en Vercel



Fuente. Autoría propia

Figura 24

Visualización del Frontend en <https://proyecto-elecciones-frontend.vercel.app>



Fuente. Autoría propia

Repositorio de la aplicación:

Enlace al repositorio en GitHub: <https://github.com/jwguerrero/SISPRECAS-Soft>

Manejo de operaciones CRUD en la aplicación

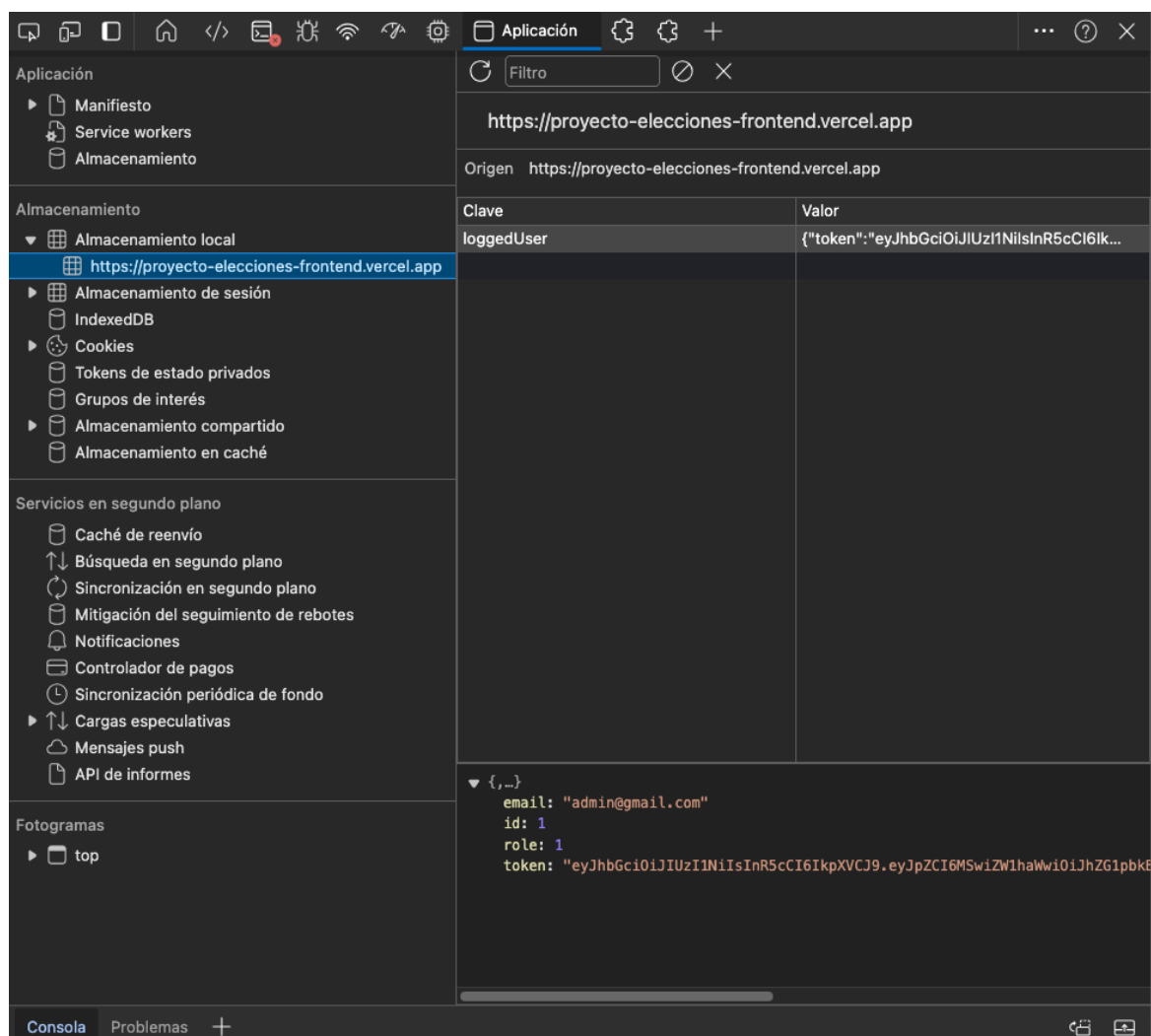
El proyecto permite la ejecución de las operaciones de registro, lectura, actualización y eliminación de usuarios desde la interfaz de usuario, de la siguiente manera:

Endpoint usuarios de la API

La API de SISPRECAS-Soft permite ejecutar operaciones en la dirección /usuarios, cuando la petición se realice por usuario autenticado y el usuario tiene el rol administrador, el cual envía el token almacenado en el navegador.

Figura 25

Visualización del token guardado en el navegador, para un usuario autenticado en el proyecto SISPRECAS-soft

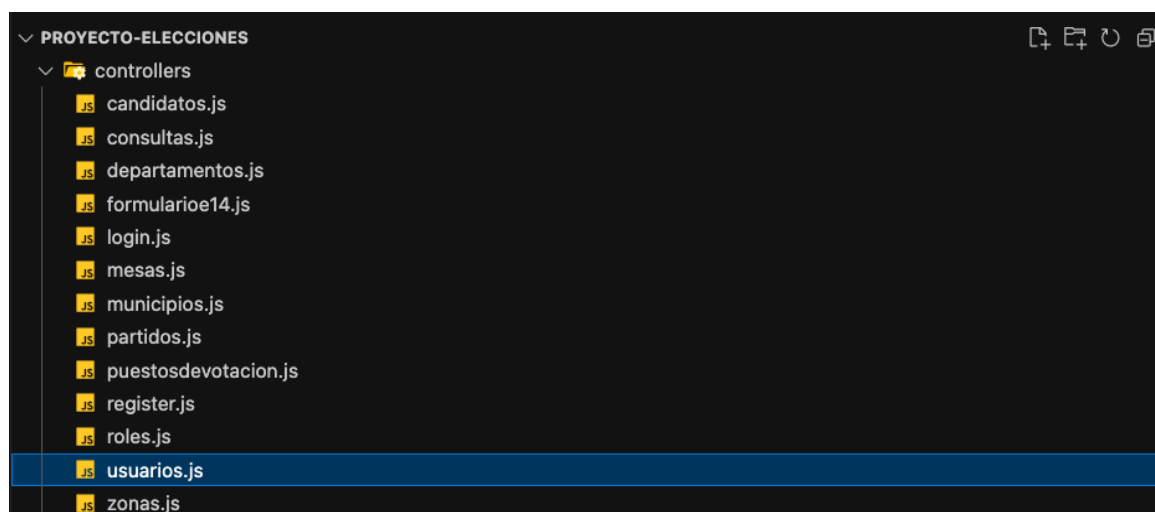


Fuente. Autoría propia

En el código del controlador de usuarios del backend podemos ver la lógica implementada para las operaciones CRUD.

Figura 26

Controlador de usuarios, usado para manejar el CRUD del endpoint usuarios



Fuente. Autoría propia

Figura 27

Función del endpoint usuarios para listar los usuarios del sistema.

Se accede al hacer una petición HTTP de tipo GET a /usuarios

Fuente. Autoría propia

```
75 usuariosRouter.get( '/usuarios', async (request, response) => {
76   try {
77     const result = await pool.query('SELECT * FROM usuarios ORDER BY id_usuario ASC;')
78     const usuarios = result.rows
79     response.json(usuarios).status(200)
80   } catch (error) {
81     response.send({ error }).status(500)
82   }
83 }
```

Figura 28

Función del endpoint usuarios, para listar los datos de un usuario del sistema, se accede al hacer una petición HTTP de tipo GET a /usuarios/:id

```

84
85 usuariosRouter.get( '/usuarios/:id', async (request, response) => {
86   const { id } = request.params
87   const result = await pool.query(`SELECT usuarios.id_usuario id_usuario, usuarios.nom_usuario nom_usuario, usuarios.
      correo_usuario correo_usuario, usuarios.clave_usuario clave_usuario, usuarios.id_role id_role, roles.nom_role nom_role,
      usuarios.id_puestodevotacion id_puestodevotacion, puestosdevotacion.nom_puestodevotacion nom_puestodevotacion, usuarios.
      url_imagen url_imagen FROM usuarios JOIN roles ON roles.id_role = usuarios.id_role JOIN puestosdevotacion ON usuarios.
      id_puestodevotacion = puestosdevotacion.id_Puestodevotacion WHERE usuarios.id_usuario = ${ id }`)
88   /* console.log(result.rows) */
89   const usuario = result.rows
90   response.json(usuario)
91 })
92

```

Fuente. Autoría propia

Figura 29

Función del endpoint usuarios, para registrar un nuevo usuario. Se accede al hacer una petición HTTP de tipo POST a /usuarios

```

92
93 usuariosRouter.post( '/usuarios', upload.single('url_imagen'), async (request, response) => {
94   /* console.log(request.body) */
95   /* const sentFile = request.file !== undefined ? request.file.path : '' */
96   let urlImagen = request.file !== undefined ? `${request.file.path}` : '/images/user-icon.png'
97
98   /* const file = request.file */
99   const { id_usuario, nom_usuario, clave_usuario, correo_usuario, id_role, id_puestodevotacion, url_imagen } = request.body
100   const usuario = await pool.query(`SELECT id_usuario FROM usuarios WHERE id_usuario = $1;`, [id_usuario])
101   /* urlImagen = (urlImagen === undefined || urlImagen === '' || urlImagen === null) ? '' : urlImagen */
102   try {
103     const { id_usuario, nom_usuario, correo_usuario, clave_usuario, id_role, id_puestodevotacion, url_imagen } = request.body
104     const saltRounds = 10;
105     const passwordHash = await bcrypt.hash(clave_usuario, saltRounds);
106     await pool.query(`INSERT INTO usuarios (id_usuario, nom_usuario, correo_usuario, clave_usuario, id_role,
      id_puestodevotacion, url_imagen) VALUES ($1, $2, $3, $4, $5, $6, $7)`, [id_usuario, nom_usuario, correo_usuario,
      passwordHash, id_role, id_puestodevotacion, urlImagen])
107     response.status(200).json(request.body)
108   } catch (error) {
109     response.send(error)
110   }
111 })

```

Fuente. Autoría propia

Figura 30

Función del endpoint usuarios, para actualizar un usuario del sistema. Se accede al hacer una petición HTTP PUT a /usuarios/:id

```

112
113 usuariosRouter.put( `/usuarios/:id`, upload.single('url_imagen'), async (request, response) => {
114   /* console.log(request.body)
115   console.log(request.file) */
116   /* const sentFile = request.file !== undefined ? request.file.path : '' */
117   /* console.log({ sentFile }) */
118   let urlImagen = request.file !== undefined ? `${request.file.path}` : ''
119   /* console.log({ urlImagen }) */
120
121   /* const file = request.file */
122   /* console.log({ file }) */
123   const id = Number(request.params.id)
124   const { nom_usuario, clave_usuario, correo_usuario, id_role, id_puestodevotacion, url_imagen } = request.body
125   const usuario = await pool.query('SELECT id_usuario FROM usuarios WHERE id_usuario = $1;', [id])
126   const imagenAnteriorTemp = await pool.query('SELECT url_imagen FROM usuarios WHERE id_usuario = $1;', [id])
127   const imagenAnterior = imagenAnteriorTemp.rows[0].url_imagen
128   /* console.log({ imagenAnterior }) */
129   /* urlImagen = (urlImagen === undefined || urlImagen === '' || urlImagen === null) ? imagenAnterior : urlImagen */
130   /* console.log(usuario.rows) */
131   const saltRounds = 10;
132   const passwordHash = await bcrypt.hash(clave_usuario, saltRounds);
133   try {
134     const query = await pool.query('UPDATE usuarios SET nom_usuario = $1, clave_usuario = $2, correo_usuario = $3,
135     id_puestodevotacion = $4, id_role = $5, url_imagen = $6 WHERE id_usuario= $7;', [nom_usuario, passwordHash, correo_usuario,
136     id_puestodevotacion, id_role, urlImagen, id])
137     if (usuario.rows.length === 0){
138       return response.json({ message: 'user not found'})
139     }
140     return response.status(200).end()
141   } catch (error) {
142     return response.json(error).status(204)
143   }
144 }
145 })
146
147

```

Fuente. Autoría propia

Figura 31

Función del endpoint usuarios, para eliminar un usuario del sistema. Se accede al hacer una petición HTTP DELETE a /usuarios/:id

```

143
144 usuariosRouter.delete( `/usuarios/:id`, async (request, response) => {
145   const { id } = request.params
146   const query = await pool.query('DELETE FROM usuarios WHERE id_usuario=$1',[id])
147   response.status(204).end()
148 }
149
150 module.exports = usuariosRouter

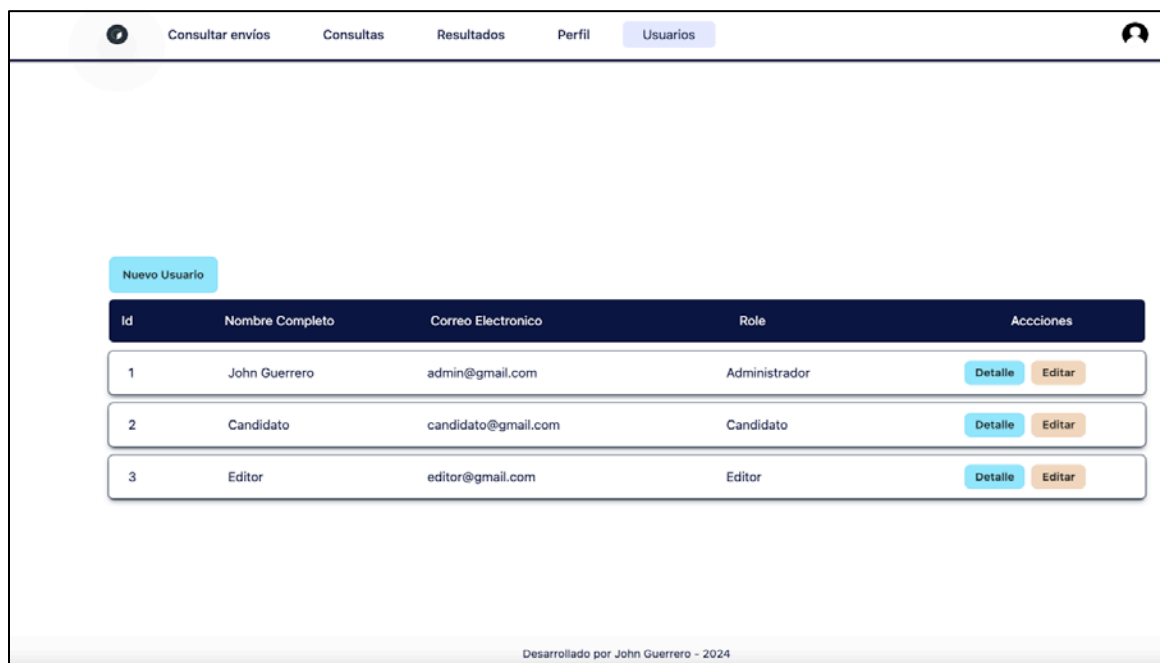
```

Fuente. Autoría propia

Estás operaciones se pueden realizar directamente desde la interfaz de usuario de SISPRECAS-Soft, por el usuario administrador, para esto debe ingresar al menú usuarios.

Figura 32

Vista del menú usuarios disponible para el usuario administrador



Id	Nombre Completo	Correo Electronico	Role	Acciones
1	John Guerrero	admin@gmail.com	Administrador	Detalle Editar
2	Candidato	candidato@gmail.com	Candidato	Detalle Editar
3	Editor	editor@gmail.com	Editor	Detalle Editar

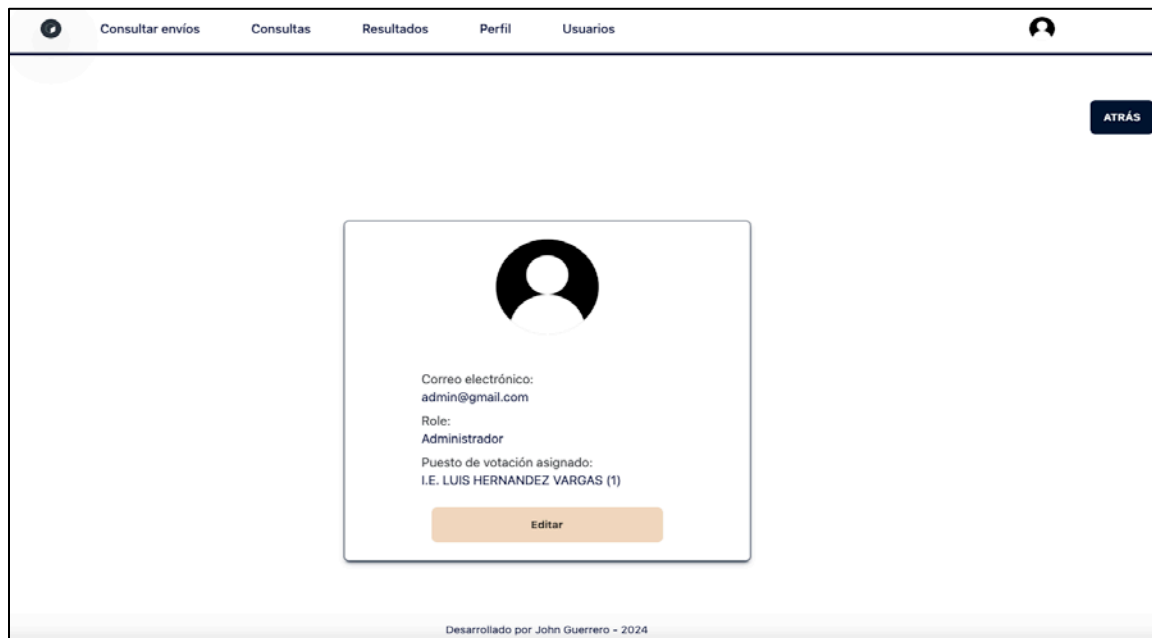
Desarrollado por John Guerrero - 2024

Fuente. Autoría propia

Al seleccionar la opción “Detalle” mostrará la información del usuario:

Figura 33

Visualización del detalle de usuario desde el perfil del administrador

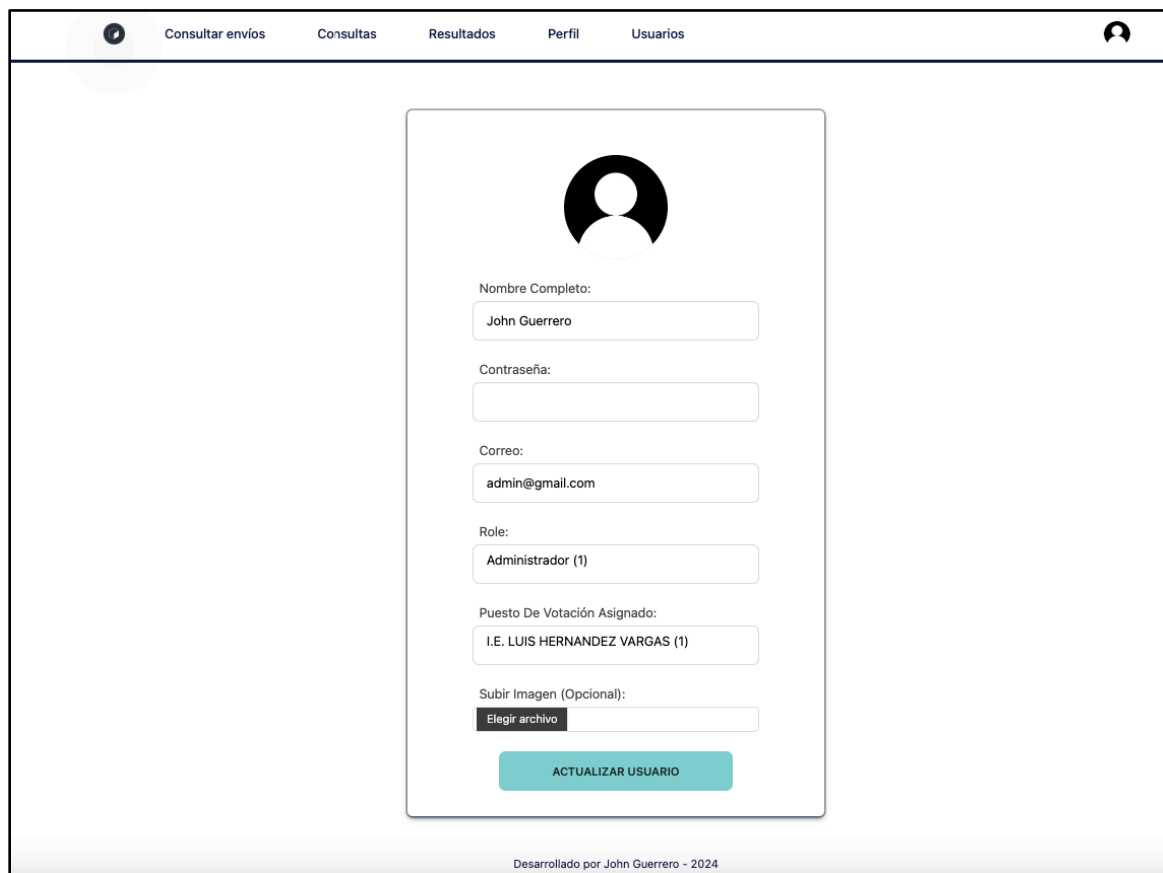


Fuente. Autoría propia

Al seleccionar la opción “Editar” podrá actualizar los datos del usuario

Figura 34

Visualización de la actualización de usuario desde el usuario administrador



The screenshot displays a web interface for updating a user. At the top, there is a navigation bar with the following items: 'Consultar envíos', 'Consultas', 'Resultados', 'Perfil', and 'Usuarios'. A user profile icon is visible in the top right corner. The main content area features a central form with a placeholder for a user profile picture. Below the picture, the form contains the following fields and labels:

- Nombre Completo: John Guerrero
- Contraseña: (empty field)
- Correo: admin@gmail.com
- Role: Administrador (1)
- Puesto De Votación Asignado: I.E. LUIS HERNANDEZ VARGAS (1)
- Subir Imagen (Opcional): Elegir archivo

At the bottom of the form is a teal button labeled 'ACTUALIZAR USUARIO'. Below the form, the footer text reads 'Desarrollado por John Guerrero - 2024'.

Fuente. Autoría propia

Al seleccionar la opción “Nuevo usuario” en la lista de usuarios podrá registrar un nuevo administrador, candidato o editor”.

Figura 35

Visualización de la edición de usuario desde el usuario administrador

The screenshot displays a web application interface for user management. At the top, a navigation bar includes links for 'Consultar envíos', 'Consultas', 'Resultados', 'Perfil', and 'Usuarios', along with a user profile icon. A dark 'ATRÁS' button is positioned in the top right corner. The main content area features a central form for adding a user, with the following fields and options:

- Nombre Completo:** A text input field.
- Contraseña:** A password input field.
- Correo:** An email input field.
- Role:** A dropdown menu currently showing 'Administrador (1)'.
- Puesto De Votación Asignado:** A dropdown menu currently showing 'I.E. LUIS HERNANDEZ VARGAS (1)'.
- Subir Imagen (Opcional):** A file upload area with an 'Elegir archivo' button.

Below the form is a teal 'AGREGAR USUARIO' button. At the bottom of the page, a footer reads 'Desarrollado por John Guerrero - 2024'.

Fuente. Autoría propia

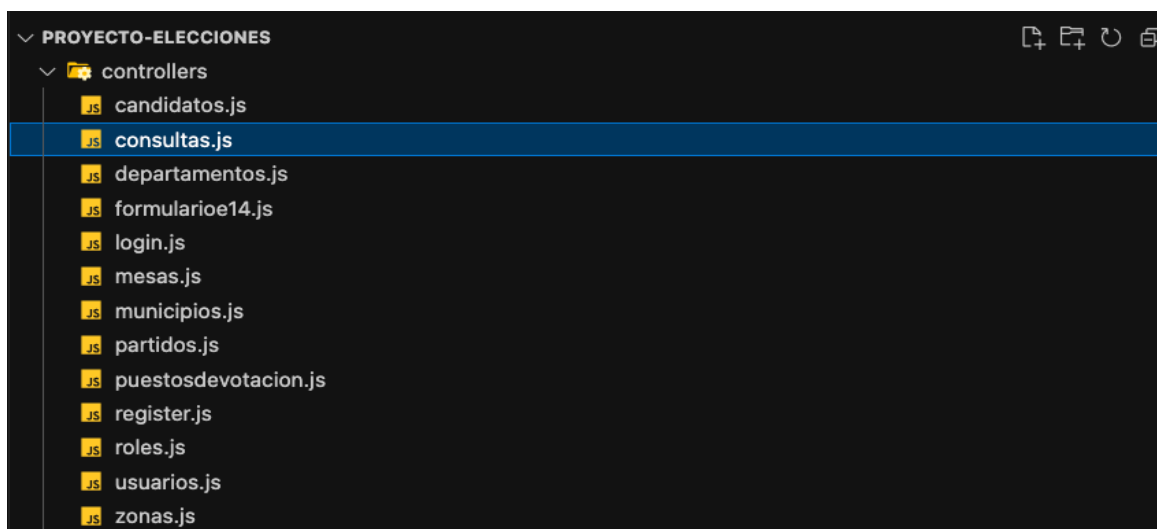
Informes estadísticos. SISPRECAS-Soft recopila los datos ingresados de las distintas mesas de votación y procesa los datos para generar un informe de los resultados. Este proceso es realizado desde en el backend y en el frontend.

Consulta de resultados en el backend

El controlador consultas tiene la responsabilidad de manejar las peticiones al endpoint /consultas, donde responde exclusivamente a la generación de resultados, usados para los informes la siguiente función:

Figura 36

Visualización de la edición de usuario desde el usuario administrador



Fuente. Autoría propia

Figura 37

Función del controlador consultas, dedicada a manejar las peticiones al endpoint/consultas/resultados/candidatos, la cual permite generar los datos consolidados de los candidatos, para la generación de informes.

```
340
341 consultasRouter.get('/consultas/resultados/candidatos', async(request, response) => {
342   try {
343     const query = await pool.query(`
344       SELECT resultados.id_candidato, candidatos.nom_candidato, candidatos.url_imagen, candidatos.id_partido, partidos.
345       nom_partido, resultados.votos
346       FROM
347         (SELECT id_candidato, sum(votos) votos from v_c
348          WHERE estado_envio_mesa = true
349          GROUP BY id_departamento, id_candidato) resultados
350       JOIN candidatos ON resultados.id_candidato = candidatos.id_candidato
351       JOIN partidos ON candidatos.id_partido = partidos.id_partido
352       GROUP BY resultados.id_candidato, candidatos.nom_candidato, candidatos.url_imagen, candidatos.id_partido, partidos.
353       nom_partido, resultados.votos
354       ORDER BY votos DESC
355     `)
356     const queryrows = query.rows
357     /* console.log({ queryrows }) */
358     if(queryrows[0]) {
359       return response.json(queryrows).status(200)
360     } else {
361       return response.json({error: 'Consulta vacia'}).status(200)
362     }
363   } catch (error){
364     return response.send({ error }).status(500)
365   }
366 }
```

Fuente. Autoría propia

Figura 38

Función del controlador consultas, dedicada a manejar las peticiones al endpoint/consultas/resultados/otras-opciones, la cual permite generar los datos consolidados de votos en blanco, votos no marcados y nulos, para la generación de informes.

```

367 consultasRouter.get('/consultas/resultados/otras-opciones', async(request, response) => {
368     try {
369         const query = await pool.query('SELECT SUM(mesas.votos_blanco) AS votos_blanco, SUM(mesas.votos_nulos) AS votos_nulos, SUM(mesas.votos_no_marcados) AS votos_no_marcados FROM mesas JOIN puestosdevotacion ON mesas.id_puestodevotacion = puestosdevotacion.id_puestodevotacion')
370         const queryrows = query.rows[0]
371         /* console.log({ queryrows }) */
372         if(queryrows.votos_blanco) {
373             return response.json(queryrows).status(200)
374         } else {
375             return response.json(objectForError).status(200)
376         }
377     } catch (error) {
378         return response.send({ error }).status(500)
379     }
380 })
381

```

Fuente. Autoría propia

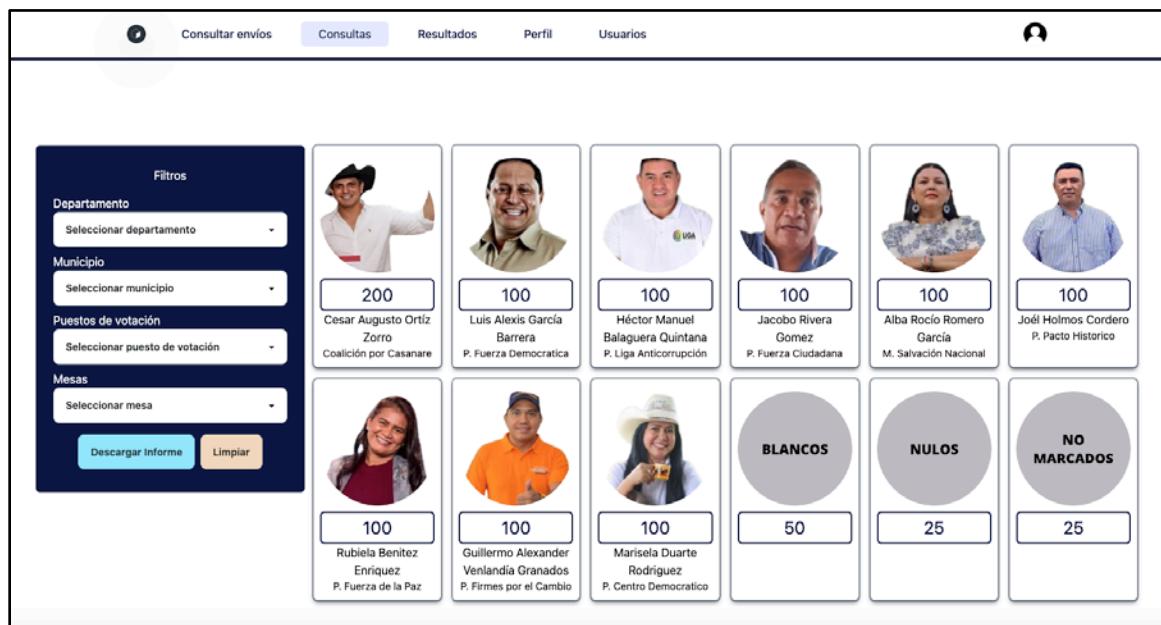
Las anteriores funciones consultas una vista en la base de datos que provee los datos de las diferentes tablas de forma que se puedan procesar de manera más rápida, ya que el motor de la base de datos devuelve los datos consolidados y solo se deben presentar adecuadamente al usuario en frontend.

Visualización de datos en el FrontEnd

Los usuarios con roles administrador y candidato, puede acceder a las consultas de los consolidados en el menú “Consultas”.

Figura 39

Visualización menú consultas para usuarios con roles administrador y candidatos



Fuente. Autoría propia

En el menú consultas es posible filtrar por departamento, aunque solo está disponible Casanare, se deja abierta la posibilidad de agregar más departamentos en una actualización.

También permite filtrar los datos por municipio, puesto de votación y número de mesa.

Además SISPRECAS-Soft permite generar un informe en pdf con los datos filtrados, esto es realizado con una librería de React llamada “React PDF”.

Figura 40

Dependencia "React PDF" en el archivo package.json del frontend realizado con React



```

4     "private": true,
5     "dependencies": {
6       "@headlessui/react": "^1.7.17",
7       "@heroicons/react": "^2.0.18",
8       "@react-pdf/renderer": "^3.1.12",
9       "@testing-library/jest-dom": "^5.17.0",
10      "@testing-library/react": "^13.4.0",
11      "@testing-library/user-event": "^13.5.0",
12      "autoprefixer": "^10.4.15",

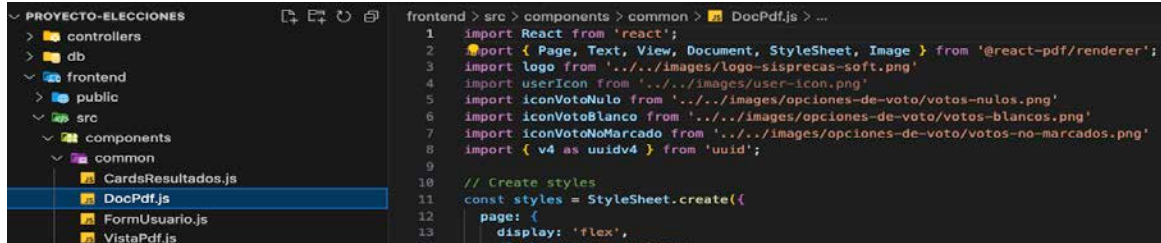
```

Fuente. Autoría propia

Está librería es utilizada para generar la vista y el documento PDF con los datos consolidados, dependiendo de los filtros usados en el menú “Consultas”

Figura 41

Layout y estilos usados para el pdf a generar



```

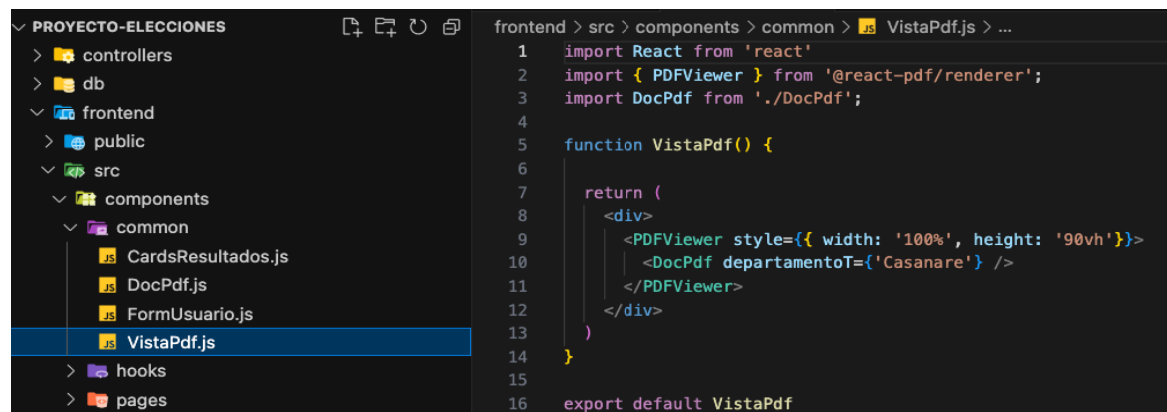
1 import React from 'react';
2 import { Page, Text, View, Document, StyleSheet, Image } from '@react-pdf/renderer';
3 import logo from '../images/logo-sisprecas-soft.png';
4 import userIcon from '../images/user-icon.png';
5 import iconVotoNulo from '../images/opciones-de-voto/votos-nulos.png';
6 import iconVotoBlanco from '../images/opciones-de-voto/votos-blancos.png';
7 import iconVotoNoMarcado from '../images/opciones-de-voto/votos-no-marcados.png';
8 import { v4 as uuidv4 } from 'uuid';
9
10 // Create styles
11 const styles = StyleSheet.create({
12   page: {
13     display: 'flex',
14     flexDirection: 'column'

```

Fuente. Autoría propia

Figura 42

Uso de la librería "React PDF" para generar el documento pdf con los datos consolidados



```
PROYECTO-ELECCIONES
├── controllers
├── db
├── frontend
├── public
├── src
│   ├── components
│   │   ├── CardsResultados.js
│   │   ├── DocPdf.js
│   │   ├── FormUsuario.js
│   │   └── VistaPdf.js
│   ├── hooks
│   └── pages
└── ...

frontend > src > components > common > VistaPdf.js > ...
1  import React from 'react'
2  import { PDFViewer } from '@react-pdf/renderer';
3  import DocPdf from './DocPdf';
4
5  function VistaPdf() {
6
7      return (
8          <div>
9              <PDFViewer style={{ width: '100%', height: '90vh'}}>
10                 <DocPdf departamentoT='Casanare' />
11             </PDFViewer>
12         </div>
13     )
14 }
15
16 export default VistaPdf
```

Fuente. Autoría propia

Plan de pruebas

El plan de pruebas se ejecutó para la validación de formularios, la verificación de la respuesta de la API y la protección de la información a través de la validación de usuarios para acceder a los recursos protegidos.

Estas verificaciones se desarrollaron creando patrones de código utilizados actualmente para prevenir ataques maliciosos a la base de datos del sistema; como es el caso de las inyecciones SQL y se complementa con la plataforma de POSTMAN, un software para validar el backend de una aplicación a través de una interfaz gráfica que permite simular las vulnerabilidades del sistema.

Figura 43

Validación de inicio de sesión

Inicio

Sisprecas Soft

Correo Electrónico
admin@gmail.com

Contraseña
.....

Email o contraseña incorrectos

ENVIAR

Desarrollado por John Guerrero - 2024

Fuente. Autoría propia

Figura 44

Validación de campos al enviar el formulario E14

Consultar envíos Consultas Resultados Perfil Usuarios

5		Alba Rocío Romero García	0
6		José Holmos Cordero	0
7		Rubiela Benitez Enriquez	0
8		Guillermo Alexander Venlandia Granados	0
9			0

Enviar formulario

Nota: Por favor verifique la información del formato E14 antes de enviarla. Recuerda que cada envío de información es responsabilidad de usted.

CANCELAR ENVIAR

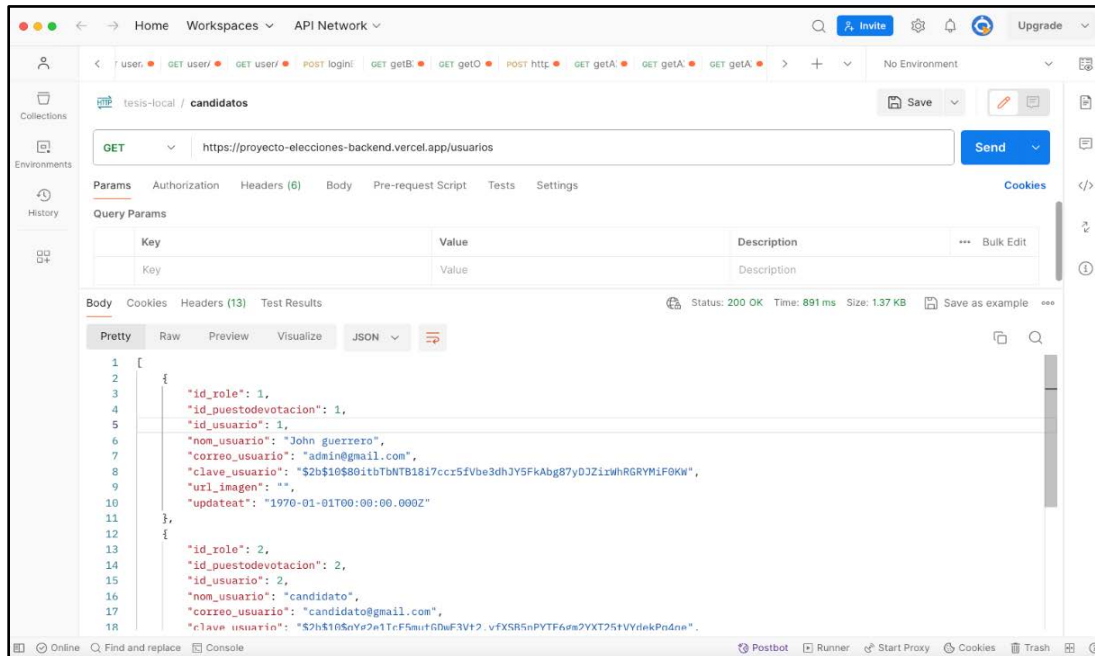
ENVIAR

Desarrollado por John Guerrero - 2024

Fuente. Autoría propia

Figura 45

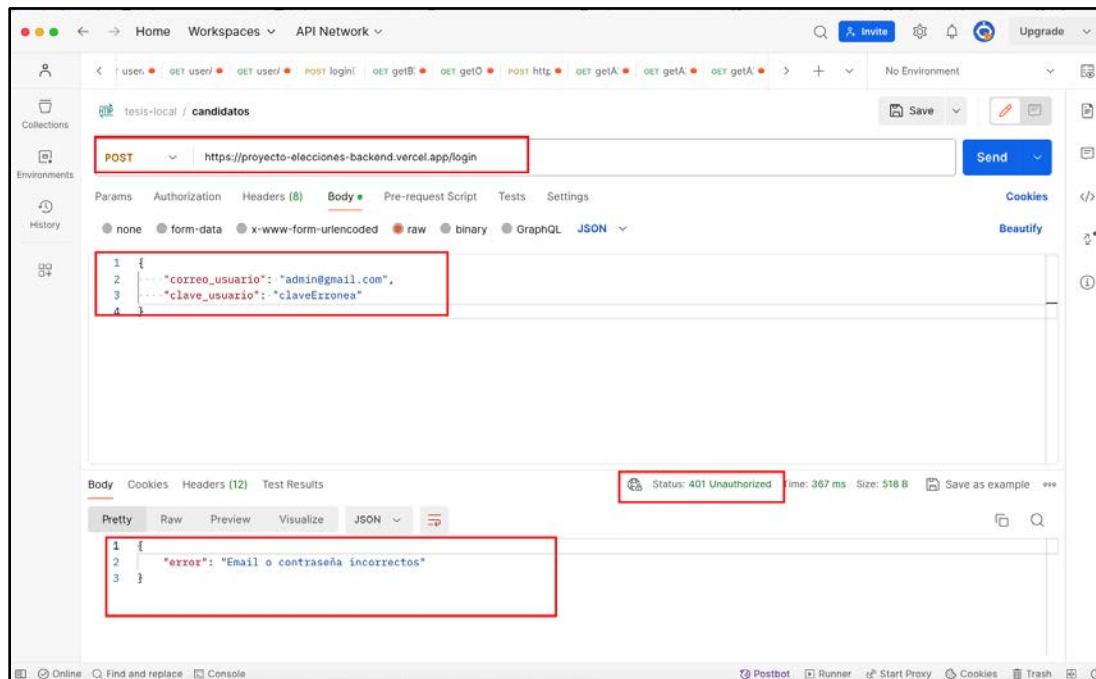
Verificación de funcionamiento de la API con Postman



Fuente. Autoría propia

Figura 46

Verificación de restricciones de acceso a personas o sistemas no autorizados



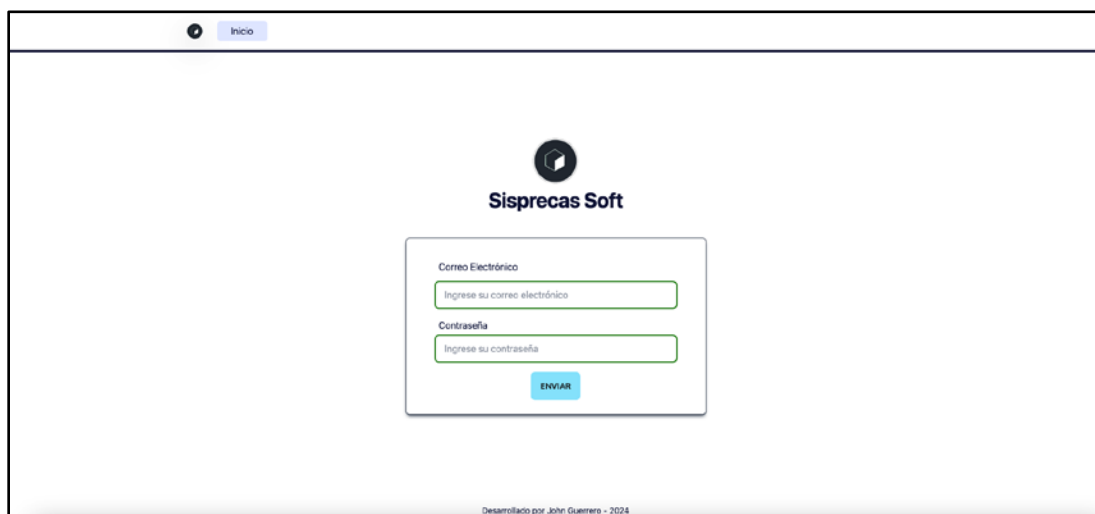
Fuente. Autoría propia

Resultados

Con la utilización de la metodología de prototipos se obtuvo una mejor presentación del diseño sin afectar la funcionalidad del sistema, lo que permitió mejorar el aspecto de la interfaz gráfica y hacerla más amigable para los usuarios.

Figura 47

Inicio de sesión



The image shows a web browser window displaying a login page. At the top left, there is a navigation bar with a home icon and the text 'Inicio'. In the center of the page, there is a logo consisting of a square with a circle inside, followed by the text 'Sisprecas Soft'. Below the logo is a login form with two input fields: 'Correo Electrónico' with the placeholder text 'Ingrese su correo electrónico', and 'Contraseña' with the placeholder text 'Ingrese su contraseña'. Below these fields is a blue button labeled 'ENVIAR'. At the bottom of the page, there is a small footer that reads 'Desarrollado por John Guerrero - 2024'.

Fuente. Autoría propia

Figura 48

Página inicial del usuario Administrador

Fuente. Autoría propia

Figura 49

Añadir información de la mesa de votación con el formulario E14-1

Departamento	Municipio	Puesto	Mesa
Casanare	Yopal	I.E. LUIS HERNANDEZ V	1
Total de sufragantes formato e11: 0			
Total de votos en la urna: 0			
Total de votos inheridos: 0			
1		Cesar Augusto Ortiz Zorio	0
2		Luis Alexis García Barrera	0
3		Héctor Manuel Balaguera Quintana	0
4		Jacobo Rivera Gomez	0
5		Alba Rocío Romero García	0
6		Johel Holmos Cordero	0
7		Rubielia Benítez Enríquez	0
8		Guillermo Alexander Venanda Granados	0
9		Marisela Duarte Rodríguez	0

Fuente. Autoría propia

Figura 50

Añadir información de la mesa de votación con el formulario E14-2

Desarrollado por John Guerrero - 2024

Fuente. Autoría propia

Figura 51

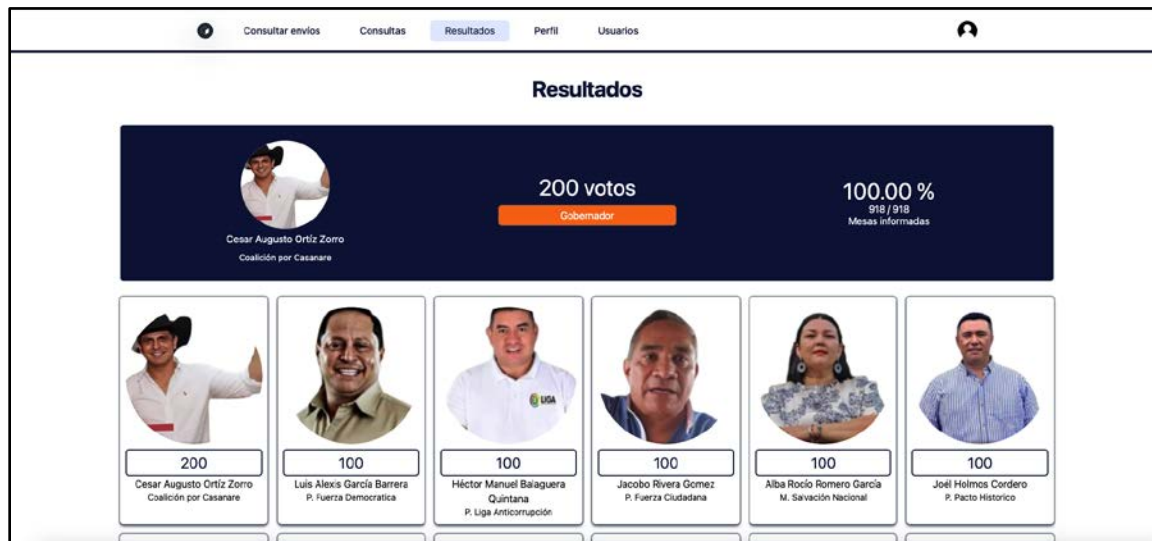
Página de Consulta con filtros por departamento, municipio, puesto de votación y mesas

Desarrollado por John Guerrero - 2024

Fuente. Autoría propia

Figura 52

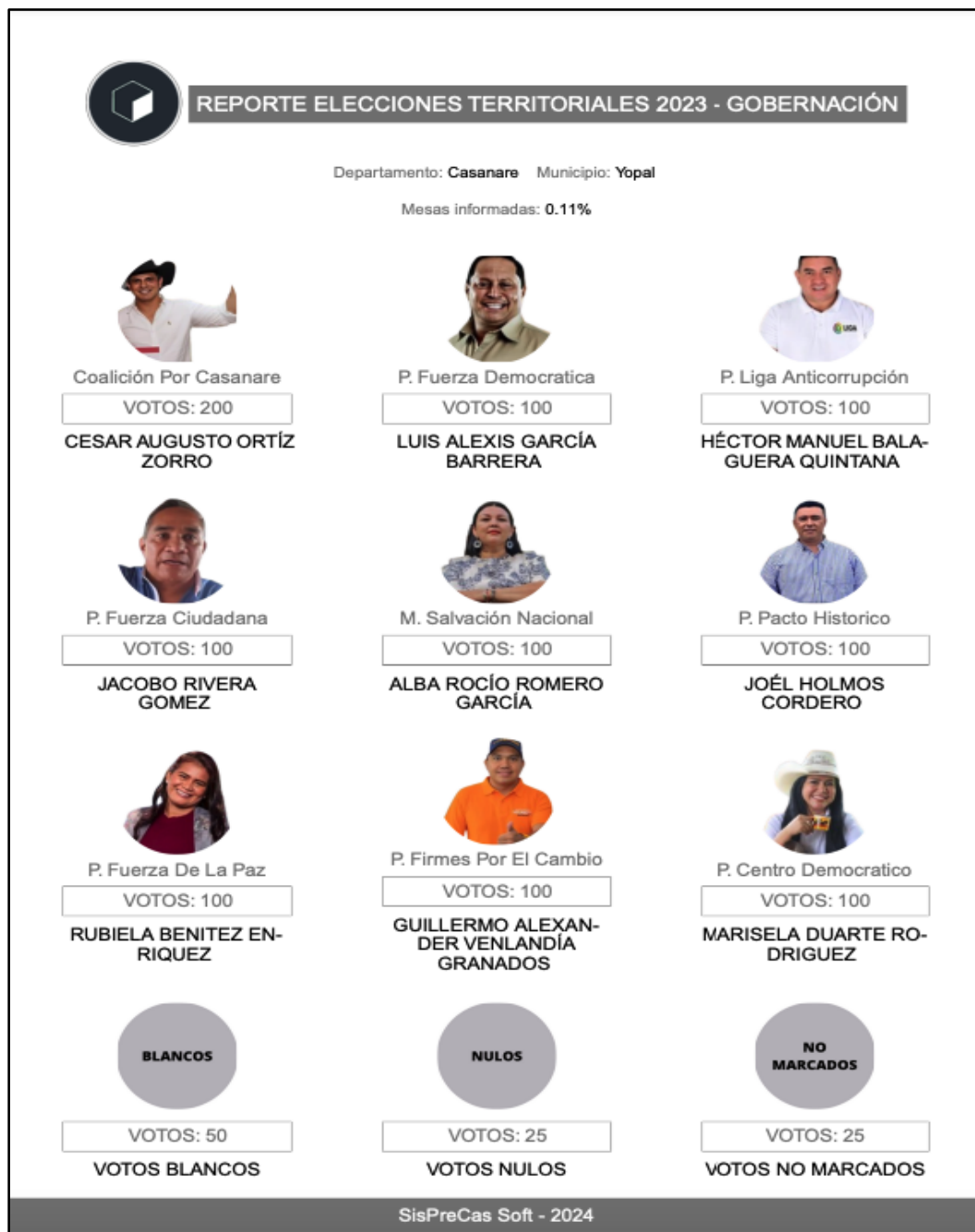
Página de resultados: Consolidado de votaciones



Fuente. Autoría propia

Figura 53

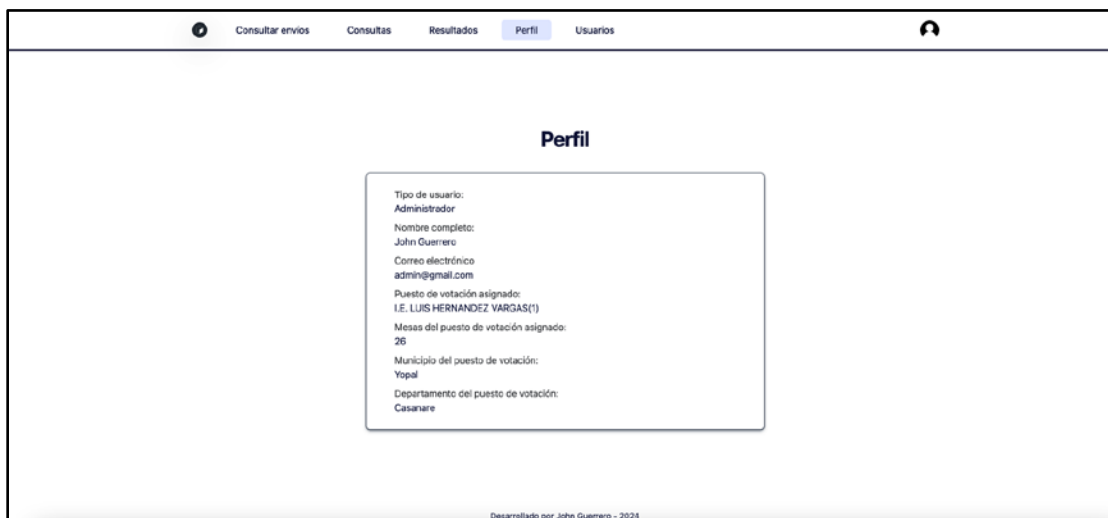
Generación de informes desde la página de consultas



Fuente. Autoría propia

Figura 54

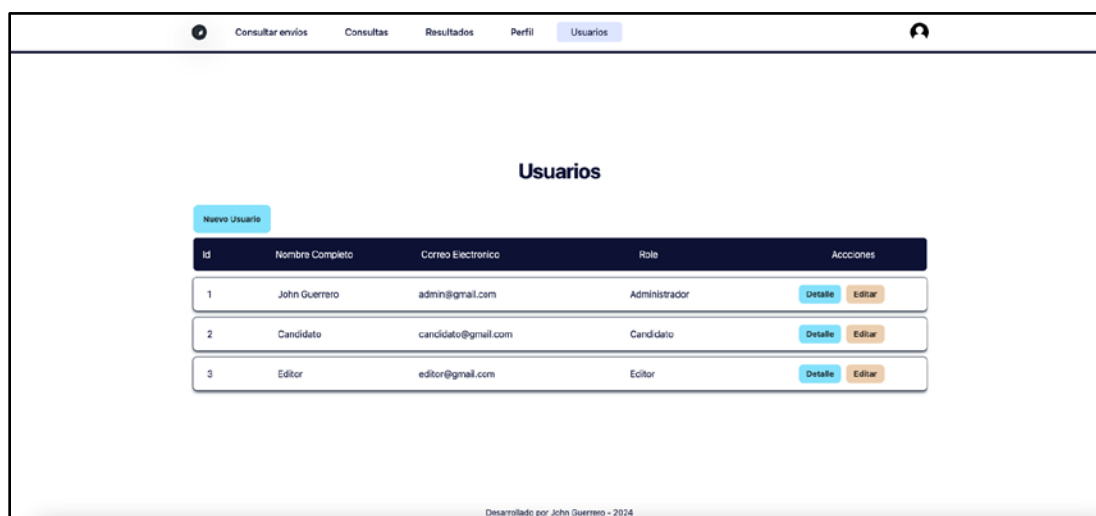
Visualización del perfil de usuario



Fuente. Autoría propia

Figura 55

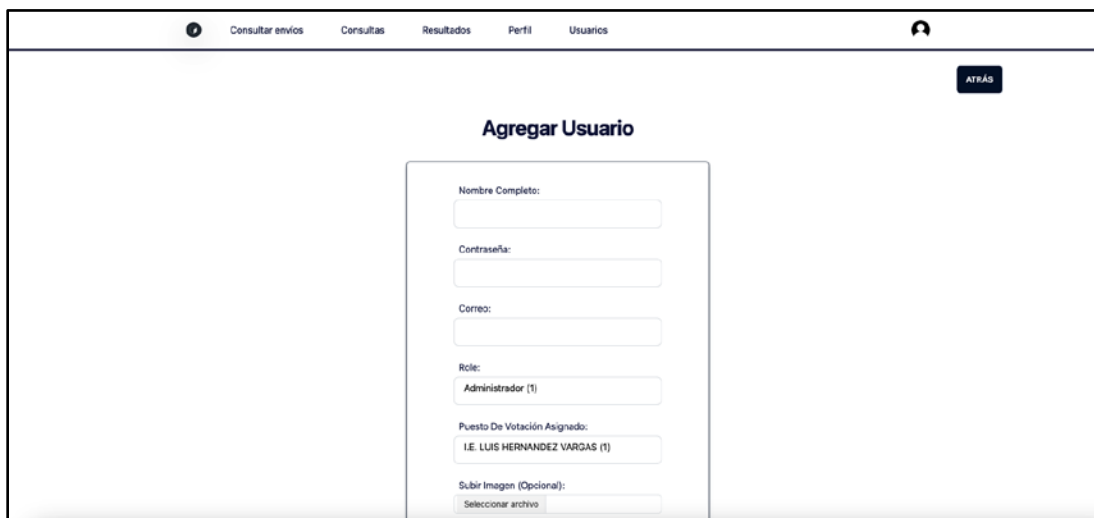
Página de gestión de usuarios



Fuente. Autoría propia

Figura 56

Agregar nuevo usuario



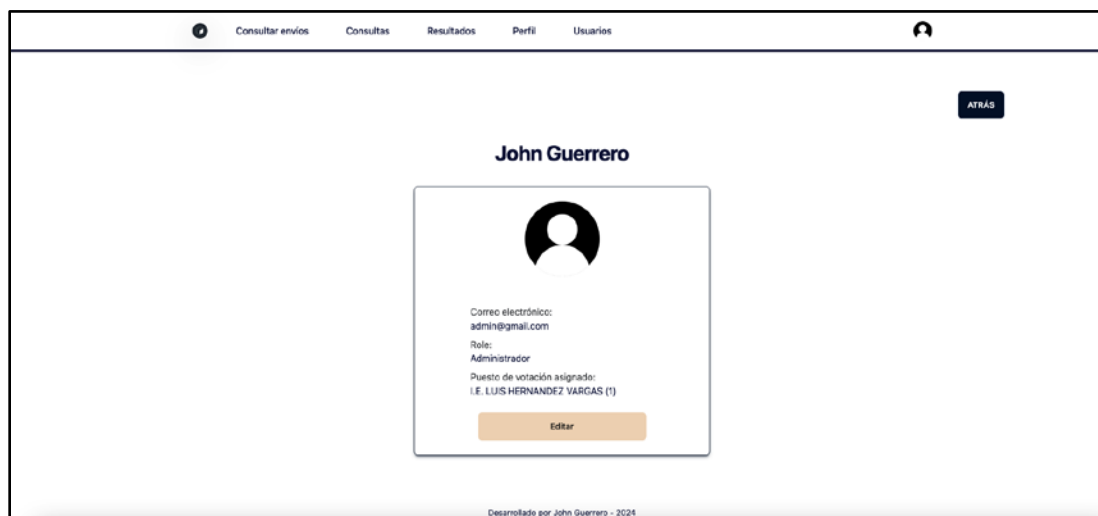
The screenshot shows a web application interface for adding a new user. At the top, there is a navigation bar with links: 'Consultar envíos', 'Consultas', 'Resultados', 'Perfil', and 'Usuarios'. A dark 'ATRÁS' button is located in the top right corner. The main content area is titled 'Agregar Usuario' and contains a form with the following fields:

- Nombre Completo:
- Contraseña:
- Correo:
- Role:
- Puesto De Votación Asignado:
- Subir Imagen (Opcional):


Fuente. Autoría propia

Figura 57

Visualización de perfil de usuario con opción de edición



The screenshot shows a user profile page for 'John Guerrero'. The navigation bar is identical to Figure 56. A dark 'ATRÁS' button is in the top right. The profile card displays:

- Nombre: **John Guerrero**
- Imagen de perfil: 
- Correo electrónico: admin@gmail.com
- Role: Administrador
- Puesto de votación asignado: I.E. LUIS HERNANDEZ VARGAS (1)
- Botón: **Editar**

At the bottom of the page, it says 'Desarrollado por John Guerrero - 2024'.


Fuente. Autoría propia

Figura 58*Edición de usuario*

Consultar envíos Consultas Resultados Perfil Usuarios

ATRÁS

Editar Usuario



Nombre Completo:
John Quintero

Contraseña:

Correo:
admin@gmail.com

Role:
Administrador (1)

Fuente. Autoría propia

Repositorio de la aplicación (Backend y FrontEnd).

A continuación se comparte el enlace a donde se encuentra ubicado los archivos de desarrollo del sistema de información para su correcto uso e instalación.

Enlace: <https://github.com/jwguerrero/SISPRECAS-Soft>

Glosario

Arquitectura Cliente/Servidor: Un enfoque en el diseño de software en el que las tareas se dividen entre los dispositivos cliente (como computadoras de usuarios) y un servidor central que proporciona servicios y recursos.

BackEnd: La parte invisible de una aplicación o sitio web que maneja las funcionalidades detrás de escena, como el procesamiento de datos, la lógica de negocio y la gestión de bases de datos.

CSS: Cascading Style Sheets (Hojas de Estilo en Cascada) es un lenguaje utilizado para dar estilo y presentación a las páginas web, controlando aspectos como colores, fuentes y diseño.

Diagramas relacionales: Representaciones gráficas que muestran cómo las tablas de una base de datos se relacionan entre sí, ayudando a comprender la estructura y las conexiones de los datos almacenados.

Diseño Web: La planificación y creación de la apariencia visual y la estructura de un sitio web para que sea atractivo y fácil de usar.

Dominio: Una dirección única en Internet que se utiliza para identificar un sitio web, como "ejemplo.com".

Formulario web: Un conjunto de campos y elementos en una página web que permiten a los usuarios ingresar y enviar información, como registros o comentarios.

FrontEnd: La parte visible de una aplicación o sitio web con la que interactúas directamente. Incluye el diseño, la presentación y la interacción de usuario, todo lo que ves y con lo que interactúas en tu navegador.

Internet: Una vasta red global de computadoras interconectadas que permite la comunicación, el intercambio de información y el acceso a recursos en todo el mundo.

Interfaz de usuario: El punto de interacción entre los usuarios y una aplicación, sistema o dispositivo, que permite la comunicación y el control.

Inicio de sesión: El proceso mediante el cual los usuarios acceden a una aplicación o sistema proporcionando sus credenciales, como nombre de usuario y contraseña.

JavaScript: Un lenguaje de programación ampliamente utilizado en el desarrollo web para crear interacciones dinámicas y en tiempo real en las páginas, como animaciones y cambios de contenido sin recargar la página.

JSON: JavaScript Object Notation (Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos que es fácil de leer para los humanos y fácil de procesar para las computadoras.

Lenguaje de programación: Un conjunto de instrucciones y reglas que permiten a los humanos comunicarse con las computadoras y desarrollar software.

Node.js: Un entorno de ejecución de JavaScript en el servidor, que permite construir aplicaciones escalables y rápidas en el lado del servidor.

Normalización de bases de datos: Un proceso de diseño en la gestión de bases de datos que busca eliminar la redundancia y garantizar la integridad de los datos a través de la organización eficiente de las tablas y sus relaciones.

Página web: Un documento en línea que puede contener texto, imágenes, enlaces y otros elementos multimedia, y que se accede a través de un navegador web.

PostgreSQL: Un sistema de gestión de bases de datos relacional (DBMS) que permite almacenar y recuperar datos de manera estructurada y eficiente.

Prototipo: Una versión preliminar de un producto o sistema que se crea con el propósito de probar ideas, funcionalidades y obtener retroalimentación antes de desarrollar la versión final.

React JS: Una biblioteca de JavaScript utilizada para construir interfaces de usuario interactivas y reactivas en aplicaciones web, simplificando la actualización y manipulación de la interfaz.

Servidor: Una computadora o sistema que proporciona recursos, servicios o datos a otras computadoras, conocidas como clientes, a través de una red.

Sistema de información: Un conjunto organizado de componentes que recopilan, almacenan y procesan datos para proporcionar información valiosa que ayuda en la toma de decisiones y el funcionamiento de una organización.

Software: Los programas, aplicaciones y sistemas informáticos que se ejecutan en una computadora para realizar tareas específicas.

SQL: Structured Query Language (Lenguaje de Consulta Estructurado) es un lenguaje utilizado para administrar y consultar bases de datos relacionales.

Tecnología: El conjunto de conocimientos, herramientas y técnicas utilizadas para crear soluciones y productos innovadores en diversas áreas.

Conclusiones

- La preocupación por la integridad de los procesos electorales en Colombia, resaltada por informes como el de la Misión de Observación Electoral (MOE), subraya la necesidad de abordar los desafíos que enfrenta el país en este aspecto. La propuesta de este proyecto para el departamento de Casanare representa un avance en la búsqueda de soluciones tecnológicas que reduzcan los riesgos identificados y promuevan elecciones más justas y equitativas.
- El diseño del Modelo Entidad Relación (MER) es esencial en el desarrollo de software, no solo para interpretar correctamente los requerimientos del sistema, sino también para tener una visión clara del alcance del proyecto.
- La elección de tecnologías modernas como React para el frontend y Node.js con Express para el BackEnd ha sido fundamental para el desarrollo del proyecto. Estas tecnologías no solo aumentan el rendimiento de un sitio web, sino que también utilizan menos recursos del servidor y permiten mostrar información de la base de datos sin recargar las páginas o subpáginas del aplicativo.
- La implementación del plan de pruebas de software es una de las etapas finales del desarrollo que permite identificar y corregir posibles vulnerabilidades del sistema. En este caso, la utilización de programas informáticos como Postman permitió ahorrar tiempo y comprender el funcionamiento del código para diagnosticar los posibles errores y corregirlos.

Referencias bibliográficas

Aguirre, S. (2020). *JSON-Vol. 1: Primeros pasos-Sintaxis-Tipos de datos json*.

<https://books.google.es/books?hl=es&lr=&id=K-fqDwAAQBAJ&oi=fnd&pg=PP1&ots=3TlxbDENU&sig=FZUHQ7CfSGegqWUkKQf0LMikvgg>

alg/bd, A. G. A. <http://hp.fciencias.unam.mx/~>, & 2012, undefined. (s/f). El modelo Entidad-Relación. *academia.edu*. Recuperado el 27 de febrero de 2024, de

<https://www.academia.edu/download/53070088/er.pdf>

Arias, Á. (2014). *Bases de Datos con MySQL: 2ª Edición normalización de ba...*

<https://books.google.es/books?hl=es&lr=&id=EojJCgAAQBAJ&oi=fnd&pg=PA13&ots=FG8aWKmS6g&sig=jagqejZh2Dc8GSZHqtFMLCLOSIU>

Cabello, A. (2015). *Implantación de aplicaciones web en entornos internet, intranet y extranet*.

IFCD0210 servidor.

<https://books.google.es/books?hl=es&lr=&id=NVYpEAAAQBAJ&oi=fnd&pg=PT3&ots=dha0SHcGXg&sig=atnjBMGkNWaxT3hnRPJ0oJstCUc>

Calume, R. (2008). *Sentencias básicas usadas en la programación de computadores lenguajes de progra...*

<https://books.google.es/books?hl=es&lr=&id=7hGfDCFu1NAC&oi=fnd&pg=PA15&ots=ydMcSMG13F&sig=MUmjZpDQv2FdyVoQtDSFSu2wFVU>

Combatir el fraude electoral ¿responsabilidad de todos o del Estado? – Red Uva. (s/f). Recuperado el 27 de febrero de 2024, de <https://reduva.org/combatar-el-fraude-electoral-responsabilidad-de-todos-o-del-estado/>

De Diseño, E., Autor, G., José, M., & Auquilla, G. (2018). *Diseño metodológico para el desarrollo de interfaces gráficas en páginas web utilizando los lenguajes HTML 5 y CSS 3 diseño web.*

<http://dspace.unach.edu.ec/handle/51000/5163>

Debrauwer, L., & Heyde, F. Van der. (2016). *UML 2.5: iniciación, ejemplos y ejercicios corregidos diagramas relacional...*

https://books.google.es/books?hl=es&lr=&id=sCU_bpeIECAC&oi=fnd&pg=PA11&ots=fJUu71p1Vn&sig=BWqOi0B8Qs0_JuMP8RF_TAaCnEs

Desarrollo De Software, M. DE, & Gabriel Pacienza, E. (s/f). *FACULTAD DE QUÍMICA E INGENIERIA “FRAY ROGELIO BACON” PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA SANTA MARIA DE LOS BUENOS AIRES Cátedra Seminario de Sistemas.*

Descubre las ventajas de la arquitectura cliente servidor en tu empresa | Actualizado febrero 2024.

(s/f). Recuperado el 26 de febrero de 2024, de <https://arquitecturanoticias.com/blog/ventajas-de-la-arquitectura-cliente-servidor/>

Díaz, T., Jersini, J., Antonio, E., Paredes, A., De, S., & Transaccionales, I. (2017). *Comparación del rendimiento entre angular4 y reactjs, basado en el modelo rail, en la progressive web app de Glup SA reactjs.* <https://repositorio.ucv.edu.pe/handle/20.500.12692/20646>

Duran Vallejo, D. S. (2018). *Sistemas de información urna electrónica para procesos operacionales de consulta electoral de la Unidad Central del Valle del Cauca.*

<https://repositorio.uceva.edu.co/handle/20.500.12993/754>

Durango, A. (2015). *Diseño Web con CSS: 2ª Edición css.*

<https://books.google.es/books?hl=es&lr=&id=bCXsCgAAQBAJ&oi=fnd&pg=PA11&ots=mKtv0MSs29&sig=xoL4Fy9XhrdFZpuL4-Gy9dBqBQw>

- Echeverry, A., Ayala, L., tecnica, C. C.-S. et, & 2008, undefined. (s/f). Introducción a la calidad de software. software. *dialnet.unirioja.es*. Recuperado el 27 de febrero de 2024, de <https://dialnet.unirioja.es/servlet/articulo?codigo=4745899>
- Florida, S., Roy Clinton Daniel, S., Agrawal, M., Brescia, P., Olivia Ocampo, C., & Olivia, C. (s/f). Capítulo 12 Introducción a HTML y al desarrollo de sitios web formulario web. *digitalcommons.usf.edu*. Recuperado el 27 de febrero de 2024, de https://digitalcommons.usf.edu/dit_tb_spa/12/
- Ginestà, M., SI]:[sn], O. M.-, & 2012, undefined. (s/f). Bases de datos en PostgreSQL postgresql. *artpatcusco.com*. Recuperado el 27 de febrero de 2024, de <https://artpatcusco.com/sis/pdf/20140527172742en.pdf>
- Gran foro por la transparencia en las elecciones 2023 - Más Contenido - ELTIEMPO.COM*. (s/f). Recuperado el 27 de febrero de 2024, de <https://www.eltiempo.com/mas-contenido/gran-foro-por-la-transparencia-en-las-elecciones-2023-805460>
- Gutiérrez Rodríguez, J. J. (s/f). *Diagramas UML de casos de uso y de requisitos*. Recuperado el 27 de febrero de 2024, de http://www.lsi.us.es/~javierj/cursos_ficheros/metricaUML/CasosUsoUML.pdf
- Haro, E., Guarda, T., ... A. P.-R. I., & 2019, undefined. (s/f). Desarrollo backend para aplicaciones web, servicios web restful: Node. js vs spring boot nodejs. *search.proquest.com*. Recuperado el 27 de febrero de 2024, de <https://search.proquest.com/openview/a78cfaa62708fd24f38ac8d1025050eb/1?pq-origsite=gscholar&cbl=1006393>

Image and Video Upload, Storage, Optimization and CDN. (s/f). Recuperado el 1 de marzo de 2024, de <https://cloudinary.com/>

Infovotantes, una herramienta tecnológica de la Registraduría Nacional al (...) - Registraduría Nacional del Estado Civil. (s/f). Recuperado el 27 de febrero de 2024, de <https://www.registraduria.gov.co/Infovotantes-una-herramienta-tecnologica-de-la-Registraduria-Nacional-al.html>

Jurados de votación: Cómo llenar formulario E-14 en elecciones presidenciales | Elecciones 2022 | Portafolio. (s/f). Recuperado el 27 de febrero de 2024, de <https://www.portafolio.co/elecciones-2022/jurados-de-votacion-como-llenar-formulario-e-14-en-elecciones-presidenciales-566095>

La Organización Electoral es propietaria del software de escrutinio para las (...) - Registraduría Nacional del Estado Civil. (s/f). Recuperado el 27 de febrero de 2024, de <https://www.registraduria.gov.co/La-Organizacion-Electoral-es-propietaria-del-software-de-escrutinio-para-las.html>

Ley 2108 de 2021 - Gestor Normativo - Función Pública. (s/f). Recuperado el 27 de febrero de 2024, de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=167946>

Mapas de Riesgo Electoral - MOE - Misión de Observación Electoral. (s/f). Recuperado el 27 de febrero de 2024, de <https://www.moe.org.co/datos-electorales/mapas-de-riesgo-electoral/>

marcha, K. R.-A.-R. tecnología en, & 2017, undefined. (s/f). Interfaz y experiencia de usuario: parámetros importantes para un diseño efectivo interfaz de usuario. *scielo.sa.cr*. Recuperado el 27 de febrero de 2024, de https://www.scielo.sa.cr/scielo.php?script=sci_arttext&pid=S0379-39822017000500049

Martínez, R. (2015). *El proceso de desarrollo de software prototipo*.

https://books.google.es/books?hl=es&lr=&id=BTTsCgAAQBAJ&oi=fnd&pg=PA33&ots=4LYuJjnl8&sig=p_t-c3xmE_FGXXKADgs6oJb4Tf-A

Maza, M. (2012). *Javascript js*.

<https://books.google.es/books?hl=es&lr=&id=3x09sewjaHIC&oi=fnd&pg=PA7&ots=YhHLJDwUzH&sig=1yngc9h9PXXsX1L4tXlhx1xaeyc>

Metodologías de Desarrollo de Sistemas de Información. (s/f).

MIGUEL, J. T. S. (2015). *Implantación de aplicaciones web en entornos internet, intranet y extranet dominio*.

<https://books.google.es/books?hl=es&lr=&id=RtESCgAAQBAJ&oi=fnd&pg=PA1&ots=2ipJ7NcF8m&sig=dDCC7OuysAb-WoFqBLzR-mnw7Uk>

Munoz, F., Reunión, L. V.-A. de la X., & 2012, undefined. (s/f). Preproceso de formularios para el análisis de seguridad de las aplicaciones web inicio de sesión. *recsi2012.mondragon.edu*.

Recuperado el 27 de febrero de 2024, de

https://recsi2012.mondragon.edu/es/programa/recsi2012_submission_28.pdf

Qué es Frontend y Backend: principal diferencia y ejemplos. (s/f). Recuperado el 26 de febrero de

2024, de <https://platzi.com/blog/que-es-frontend-y-backend/>

¿Qué es la accesibilidad? - Aprende desarrollo web | MDN. (s/f). Recuperado el 27 de febrero de

2024, de https://developer.mozilla.org/es/docs/Learn/Accessibility/What_is_accessibility

¿Qué es la seguridad de aplicaciones? | Akamai. (s/f). Recuperado el 27 de febrero de 2024, de

<https://www.akamai.com/es/glossary/what-is-app-security>

Qué es Serverless, ventajas y servicios | OpenWebinars. (s/f). Recuperado el 1 de marzo de 2024, de

<https://openwebinars.net/blog/que-es-serverless-ventajas-y-servicios/>

¿Qué es una API? - Explicación de interfaz de programación de aplicaciones - AWS. (s/f).

Recuperado el 1 de marzo de 2024, de <https://aws.amazon.com/es/what-is/api/>

¿Qué es una API de RESTful? - Explicación de API de RESTful - AWS. (s/f). Recuperado el 1 de

marzo de 2024, de <https://aws.amazon.com/es/what-is/restful-api/>

Qué es una página web, para qué sirve y tipos - GoDaddy Resources - LATAM. (s/f). Recuperado el

27 de febrero de 2024, de <https://www.godaddy.com/resources/latam/stories/pagina-web-que-es-tipos>

¿Qué pasa con el voto electrónico en Colombia? (s/f). Recuperado el 27 de febrero de 2024, de

<https://www.unisabana.edu.co/nosotros/subsitios-especiales/especial-campus-elige-bien/que-pasa-con-el-voto-electronico-en-colombia/>

Raynaud, D. (2018). *¿Qué es la tecnología? tecnología.*

https://www.academia.edu/download/56224230/tech_cubierta_int.pdf

Rivera, F. (2008). *Base de datos relacionales sql.*

https://books.google.es/books?hl=es&lr=&id=buM5rlZME-cC&oi=fnd&pg=PA9&ots=6N1LprHHOK&sig=609aHQBpwrBj_5d3PXE6ucJF4q0

Sistema de voto electrónico seguro SVES. (s/f). Recuperado el 27 de febrero de 2024, de

<https://repositorio.unicartagena.edu.co/handle/11227/2915>

T-637-01 Corte Constitucional de Colombia. (s/f). Recuperado el 27 de febrero de 2024, de

<https://www.corteconstitucional.gov.co/relatoria/2001/T-637-01.htm>

Tutorial de diagrama de clases UML | Lucidchart. (s/f). Recuperado el 27 de febrero de 2024, de

<https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml>

Valencia Cabrera Ivalencia, L. (s/f). Normalización en Bases de Datos Relacionales. *cs.us.es*.

Recuperado el 29 de febrero de 2024, de <https://www.cs.us.es/cursos/bd-2022/temas/BD-Tema-6.pdf>

Vercel: Build and deploy the best Web experiences with The Frontend Cloud – Vercel. (s/f).

Recuperado el 1 de marzo de 2024, de <https://vercel.com/>

web, D. C.-L. de información específica en la, & 2005, undefined. (s/f). ¿Qué es internet.

books.google.com. Recuperado el 26 de febrero de 2024, de

<https://books.google.es/books?hl=es&lr=&id=9fsgpKpvKuwC&oi=fnd&pg=PA2&ots=vJYnHwcOgD&sig=Maxb1OB-xh1sIPNrQrOEaMXIzAM>