



Proyecto de Grado

**Crear un Sistema Automatizado para la Sala Situacional de la Unidad
Democrática que Permita el Monitoreo del Proceso Electoral del Estado
Trujillo.**

www.bdigital.ula.ve

Por

Bachiller: Eduardo Peñaloza

Tutor: MsC. Manuel Correa

©2021 Universidad de Los Andes Mérida, Venezuela

C.C. Reconocimiento

Br. Eduardo José Peñaloza Olmos.

Propuesta de Proyecto de Grado – Sistemas Computacionales, N° paginas 96.

Resumen

El presente proyecto de grado está orientado a automatizar la recolección de data en los procesos electorales, siendo el caso de estudio la Sala Situacional de los partidos políticos de la oposición venezolana del Estado Trujillo, la cual maneja un procedimiento de recopilación de datos de manera manual, la transmisión de la data la realiza el monitor por mensaje de texto o llamada telefónica, la misma es tomada por el receptor del circuito en la sala situacional y entregada al transcriptor en una hoja para que la procese en una hoja de cálculo, siendo esta tarea ineficiente y con la probabilidad de errores en los datos, aunado a un elevado tiempo de procesamiento que dificulta la toma de decisiones y estrategias para mejorar la participación electoral del estado. Ante esta situación, se presenta la necesidad de contar con una herramienta tecnológica que permita llevar a cabo el envío y recepción de datos de los centros electorales de manera rápida y eficiente, para que estos sean analizados con la menor brevedad posible y se pueda sacar conclusiones, toma de decisiones y recomendaciones en el mínimo tiempo admisible para mejorar la participación del electorado y los números del candidato o los candidatos de los partidos políticos de la oposición venezolana.

Palabras Clave: Sala Situacional, Automatización, Elecciones, Herramientas Tecnológicas, Partidos Políticos.

Índice

Capítulo 1	11
Introducción	11
Planteamiento Del Problema	13
1.1. Definición del Problema	13
1.2. Justificación.....	15
1.3. Alcance	16
1.4. Objetivos Del Estudio	16
1.4.1. Objetivo General	16
1.4.2. Objetivos Específicos	17
1.5. Antecedentes	17
CAPÍTULO 2	20
Marco Teórico	20
2.1. Sala Situacional	20
2.2. Partidos Políticos.....	23
2.3. Funciones De Los Partidos Políticos	24
2.4. Características De Los Partidos Políticos (Organización).....	24
2.5. Ideología.	25
2.6. Poder Político.....	25
2.7. Cargos Públicos Electivos.....	25
2.8. Monitoreo Del Proceso Electoral	26
2.9. Política Contemporánea De Venezuela	27
2.10. Estructura De La División Partidista En Venezuela.....	28
2.11. SMS (Short Message Service).....	29
2.12. Python	30
2.13. SL4A	31
2.14. Aplicación Qpython	32
2.15. Modelo Cliente-Servidor.....	33
2.16. XML-RPC	34
2.17. Servidor Web	36
2.18. Apache	36
2.19. Php	37
2.20. PostgreSQL.....	38

2.21. Modelo Vista Controlador (MVC).....	39
CAPÍTULO 3	40
Marco Metodológico	40
3.1. Fase 1.- Requerimientos.....	42
3.2. Fase 2.- Definición de las tareas y las iteraciones	42
3.3. Fase 3.- Diseño de los incrementos.....	42
3.4. Fase 4.- Desarrollo del incremento	42
3.5. Fase 5.- Validación de los incrementos	43
3.6. Fase 6.- Integración de los incrementos.....	43
3.7. Fase 7.- Entrega del producto.....	43
CAPÍTULO 4	44
Diseño Del Sistema	44
4.1. Descripción de actores	45
4.2. Requerimientos.....	46
4.2.1. Requerimientos de Entorno.....	47
4.2.2. Requerimientos Funcionales.....	48
4.2.3. Requerimientos no Funcionales.....	48
4.3. Desarrollo de la Aplicación	49
4.4. Plataforma De Comunicaciones.....	49
4.5. Plataforma de hardware.....	52
4.5.1. Hardware Aplicación Móvil.....	52
4.5.2. Hardware Del Servidor.....	52
4.6. Software de Desarrollo.....	53
4.6.1. Dispositivo Móvil (Smart phone).....	53
4.6.2. Computador (Servidor Web).....	56
4.7. Base de Datos	61
4.8. Implementación del Sistema	64
CAPÍTULO 5	70
5.1. Pruebas de caja blanca	70
5.2. Pruebas de caja negra	76
5.3. Usuario Supervisor Estatal.....	85
5.4. Usuario Supervisor Municipal.....	86
5.5. Mantenimiento	88

CAPÍTULO 6	89
Conclusiones.....	89
Recomendaciones.....	90
Referencias Bibliográficas	92

Índice de Tablas

Tabla 1. Actores que cooperan en los centros de votación.	45
Tabla 2. Actores que laboran en la sala situacional.....	46

Índice de Figuras

Figura 1 . Operatividad de procesamiento actual de recolección de data de los centros electorales.	14
Figura 2. Etapas observadas en la transmisión/recepción y procesamiento de data de un evento electoral.....	42
Figura 3. Captura pantalla etapa envió SMS.	50
Figura 4. Captura pantalla ejemplo envió SMS.	51
Figura 5. Aplicación Activa con su respectivo código a ejecutar.	53
Figura 6. Librerías Python.....	54
Figura 7. Envía la petición de comunicación con el servidor.....	54
Figura 8. Recorrido del buzón de mensaje de texto.	54
Figura 9. Envío de los data.	55
Figura 10. Código en Ejecución.....	56
Figura 11. Comienza el recorrido de la codificación.	56
Figura 12. Asignan los datos en variable Php.	57
Figura 13. Establece conexión con la Base de Datos.	57
Figura 14. Protocolo de Verificación.....	58
Figura 15. Proceso de insertar los datos como Súper-Monitor.....	58
Figura 16. Proceso de insertar los datos como Monitor.	59
Figura 17. Tabla de Monitores.....	59
Figura 18. Tabla Reportes 1.....	60
Figura 19. Tabla Reportes 2.....	60
Figura 20. Tabla Reportes 3.....	60
Figura 21. Tabla Centros.	61
Figura 22. Tabla Estados.....	62
Figura 23. Tabla Municipios.	62
Figura 24. Tabla Parroquias.....	62
Figura 25. Tabla Monitores.	63
Figura 26. Tabla Reportes.	63
Figura 27. Tabla Votantes.....	64

Figura 28. Ventana de Inicio.....	65
Figura 29. Ventana de Inicio de Registro.	66
Figura 30. Ventana de Registro.	67
Figura 31. Ventana Registro Usuario Administrador.	68
Figura 32. Ventana Crear Un Usuario.	68
Figura 33. Inicio de Sesión.....	71
Figura 34. Ventana Principal.	71
Figura 35. Botón Robot.	72
Figura 36. Limpiado de Tablas.	73
Figura 37. Simulación Iniciada.	73
Figura 38. Tabla Centros.	74
Figura 39. Simulación Municipios.....	74
Figura 40. Tabla Municipios.	75
Figura 41. Inicio Sesión Administrador.	76
Figura 42. Botón Reporte General.	77
Figura 43. Ventana Reporte General.	77
Figura 44. Por Municipios.....	78
Figura 45. Ventana Reporte Por Municipio.....	78
Figura 46. Por Parroquia.	79
Figura 47. Ventana Reporte Por Parroquia.	79
Figura 48. Por Centro.	80
Figura 49. Ventana Reporte Por Centro De Votación.....	80
Figura 50. Botón Manual.....	81
Figura 51. Ventana Botón Manual.....	81
Figura 52. Consulta Reporte General.....	82
Figura 53. Gráfica de Torta Reporte General.	82
Figura 54. Reporte General (Trujillo) Primera iteración Grafica de Torta.....	83
Figura 55. Reporte Por Municipio (Pampanito) Gráfica de Barras.	83
Figura 56. Reporte Por Parroquia (La Concepción) Grafica Torta+Barras.....	84
Figura 57. Reporte Por Centro (Código 54650).	84
Figura 58. Usuario Supervisor Estatal.	85
Figura 59. Usuario Supervisor Municipal.	86
Figura 60. Ventana 1 Usuario Supervisor Municipal.....	86
Figura 61. Ventana 2 Usuario Supervisor Municipal.....	87
Figura 62. Ventana 3 Usuario Supervisor Municipal.....	88

Capítulo 1

Introducción

En la actualidad los procesos electorales representan el mecanismo de selección de la opción de mayor aceptación popular en un territorio limitado, ya sea parroquia, municipio, estado o país. Es la forma legal para discutir y disputar en un terreno político, las diferencias ideológicas y de principios de diferentes partidos.

Son los partidos políticos quienes establecen la contienda electoral y donde muchas veces los ciudadanos creen no participar en la misma hasta que se instala el proceso, una creencia alejada de la realidad, ya que en el quehacer diario se discuten y analiza la participación de los actores políticos en la comunidad.

Desde el instante que se presentan los actores políticos, los grupos inician un proceso de recolección de data para realizar análisis estadístico de la aceptación y valorar una posible candidatura de los mismos. Ya postulado entra en juego la búsqueda de votos para lograr el objetivo de ganar.

Esta tarea a simple vista se ve sencilla, pero tiene un trabajo fuerte. En Venezuela, los procesos electorales son planificados y ejecutados por el Poder Electoral, quien lo dirige el Consejo Nacional Electoral (CNE), estableciendo las reglas de juego, centros y patrones electorales. Cada organización o grupo político tiene conocimiento de cada uno de estos elementos y busca sacar provecho de ellos, organizándose de la mejor manera para captar la mayor cantidad de votantes.

La estrategia empleada en agrupaciones es divide y vencerás, creando maquinarias sectoriales para el reclutamiento y seguimiento de electores, basándose en los elementos citados anteriormente, centros y patrones electorales. Esas maquinarias serán controladas y monitoreadas por un ente central, al cual llama Sala Situacional, según Bergonzoli (2006) establece que una Sala Situacional constituye un

“espacio virtual y matricial, de convergencia donde se conjugan diferentes saberes para la identificación y estudio de situaciones, respondiendo a inquietudes e interrogantes que tengan que ver con la articulación entre las aspiraciones y las demandas priorizadas de la población frente a los recursos disponibles” y una de sus funciones es la captura y procesamiento de datos durante el proceso electoral con el objetivo de determinar acciones basadas en los números generados comparados con las proyecciones determinadas. Los números son procesados a través de hojas de cálculo con entrada manual de los datos, esto origina que en procesos grandes los tiempos de respuestas sean lentos y la toma de decisiones extemporáneas ocasionando una posible derrota de su candidato.

Ahora bien, se han hecho múltiples estudios para la implementación de tecnologías informáticas con la finalidad de optimizar y agilizar el análisis de los procesos electorales, y la Sala Situacional no se escapa de los mismo, ya que tiene como objetivo primordial fortalecer la capacidad de respuesta para mejorar las condiciones en cada proceso con un ejercicio riguroso de recolección, priorización, análisis de la información, y generación de acciones para un mejor desempeño electoral. En vista de mejorar la operatividad de las Salas Situacionales se presenta la propuesta de diseño de un Sistema Automático de captura y procesamiento de datos para dicha sala que lo presentamos en este documento bajo los siguientes capítulos:

- El primer capítulo está destinado al planteamiento del problema de investigación. Allí se incluyen la justificación, alcances, objetivos y antecedentes.
- En el segundo capítulo se presentan los aspectos teóricos sobre los cuales se fundamenta el sistema automático de captura y procesamiento de datos para una sala situacional.
- El tercer capítulo abarca el marco metodológico implementado para el desarrollo del producto.
- El cuarto capítulo describe el desarrollo mediante la metodología Desarrollo iterativo e incremental.
- El quinto capítulo exhibe los resultados arrojados.

- Por último el capítulo sexto se refiere a las conclusiones que el autor ha realizado a partir de todos los objetivos cumplidos.

Planteamiento Del Problema

1.1. Definición del Problema

En los procesos electorales los entes participantes se activan en pro de monitorear todo el proceso, ya sea independientemente u organizados de acuerdo a su tendencia política. En Venezuela, en los últimos procesos los partidos políticos se organizan de acuerdo a su tendencia ideológica en partidos de Izquierda (Oficialismo) y partidos de Derecha (Oposición).

Para realizar el monitoreo se establecen Salas Situacionales, una central en la Ciudad de Caracas que toma la data de las Salas Regionales, realiza estimaciones, decisiones generales y las Salas Situacionales Regionales que toman la data de cada Estado correspondiente. Particularmente en el Estado Trujillo los grupos de partidos políticos de oposición establecen la Sala Situacional en la Ciudad de Valera, en dicha sala se monitorea todo el proceso electoral dividiendo las tareas de acuerdo a las acciones que se puedan presentar, por ejemplo: llegada el material electoral, instalación y apertura de los centros y mesas de votación, asignación de operadores y testigos de la mesas de votación, eventualidades presentadas, flujo de votantes, cierre de las mesas, escrutinios, totalización de los votos y entregas de las actas.

El monitoreo de flujo de votantes se realiza a partir de cierta hora de apertura de las mesas de votación y se establecen bloques para enviar la data de las mesas a la sede regional de la Sala Situacional, antes de ello se realiza el reclutamiento del personal a trabajar en tarea, creando un proceso para el envío y recepción de la data. Por cada mesa electoral y de acuerdo al número de votantes se establece una persona, llamada monitor, que se encarga de observar y contabilizar el número de electores (votantes) en cierto periodo de tiempo, una vez concluido el periodo debe notificar ese

número a una persona ubicada en la Sala Situacional, llamado receptor quien es el responsable de varias mesas electorales, se encarga de tomar la data y llevarla transcrita en un formato al operador que se encuentra en un equipo procesando la misma en una hoja de cálculo electrónica (Excel) que cuenta con las estimaciones estadísticas necesarias para la toma de decisiones (ver figura 1).

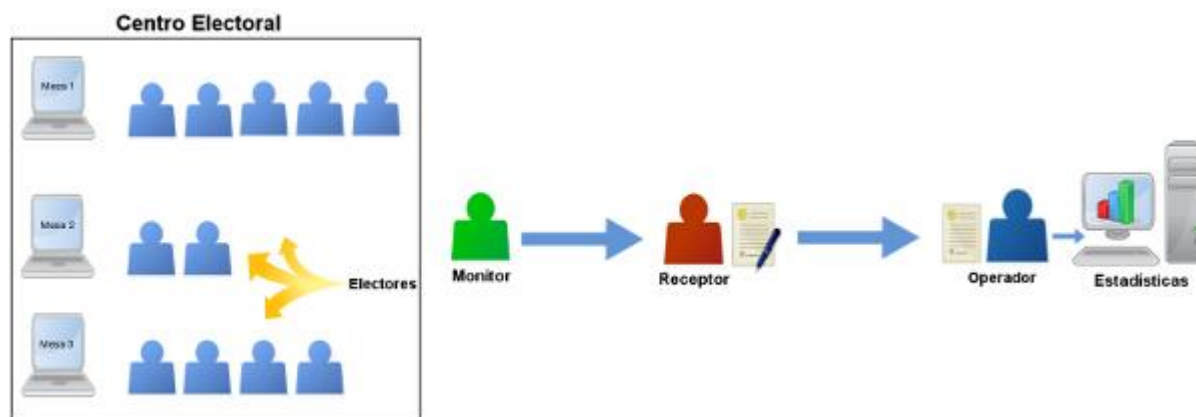


Figura 1 . Operatividad de procesamiento actual de recolección de data de los centros electorales.

Durante este proceso se presenta diversas eventualidades, en primer lugar existen centros electorales distantes que no cuentan con ningún tipo de servicio tecnológico (telefonía, Internet), esto origina que no se cuente con esa data a tiempo, el segundo es que en los bloques de data, los receptores obtienen la información de los monitores ya sea vía telefónica (llamada), mensaje texto o mensaje Whatsapp, y debe pasar la misma al operador, punto de congestionamiento por la gran cantidad de data en tan corto periodo de tiempo, de ahí , que la transcripción se vuelve lenta y el procesamiento de datos puede tardar cierta cantidad de tiempo y en algunos casos se solapan los bloques de recepción de data, originando que no se evalúe a tiempo los procesos estadísticos y en consecuencia no se tomen las decisiones con celeridad ocasionando que el candidato pueda perder la justa electoral.

De acuerdo a la situación expuesta anteriormente se plantean las siguientes interrogantes:

¿Cómo es la situación actual con relación al monitoreo del flujo de votantes a los centros electorales? ¿Existe una herramienta y/o aplicación que permita disminuir el tiempo de procesamiento de datos para su análisis estadístico? ¿Qué otras herramientas se pueden implementar para disminuir los tiempo de recepción de data y procesamiento de las mismas? ¿Cómo podemos evaluar que las data es procesada en un menor tiempo?

1.2. Justificación

Las elecciones para la vida política venezolana fomentan el interés tanto del ciudadano como de las organizaciones políticas en conocer las tendencias de la opinión pública, cuestión que se tecnifica mediante la cada vez mayor relevancia de las Salas Situacionales para monitoreo de procesos electorales; la Sala Situacional como un “espacio virtual y matricial, de convergencia donde se conjugan diferentes saberes para la identificación y estudio de situaciones” (Bergonzoli, 2006), y que en la práctica convoca expertos para la elaboración de diagnósticos y perfiles situacionales como una herramienta de apoyo inmediato, para resolver las dudas y problemas que desorientan los cursos de las decisiones al interior de las organizaciones políticas.

En consecuencia, la investigación tiene como objetivo principal crear un sistema automatizado para la Sala Situacional de la oposición en el Estado Trujillo, que reemplace el actual modelo manual aplicado en el monitoreo del proceso electoral en el Estado.

En tal sentido, se ha observado que en varios procesos electorales, la toma de decisiones en tiempo real ha sido ineficiente, donde el principal problema son las instrucciones que se les debe dar a las maquinaria de los partidos políticos para la movilización de electores, ya que el manejo manual de la data, su procesamiento y análisis es muy lento debido a que los periodos de observación y conteo establecidos son los mismos para todos los centros electorales y el envío de la data es de manera concurrente, ocasionado un cuello de botella en el operador del sistema al momento de transcribir y

adicionalmente algunos datos errados que generan incongruencias al momento de medir la proyección de votos, dando como resultado una posible derrota del candidato.

En virtud a este problema se busca un sistema que permita tomar decisiones acordes, en tiempo de ejecución para buscar estrategias y mejorar la participación del candidato de la oposición en los procesos electorales.

1.3. Alcance

Crear un sistema automatizado para la sala situacional de la Unidad Democrática, que facilite la recolección de la data, envío, recepción y procesamiento de la misma en cortos periodos de tiempo de forma eficaz, comprensible y eficiente, con la menor cantidad de errores en los centros electorales, facilitando el trabajo a los miembros de la sala situacional permitiendo la toma de decisiones y conclusiones más viables, justo en el momento preciso y oportuno. En tal sentido, esta herramienta tecnológica se encuentra orientada, hacia el perfeccionamiento y mejoramiento de la participación de candidatos en los partidos políticos de la oposición venezolana. De igual manera, establece un sistema automatizado de gestión de contenido, en todo lo referente al sistema tanto la parte de recolección de datos como la consulta de los mismos en la aplicación web, sin necesidad de un personal técnico que explique los resultados obtenidos

1.4. Objetivos Del Estudio

1.4.1. Objetivo General

Crear un sistema automatizado para la Sala Situacional de la Mesa de la Unidad Democrática o Partidos Políticos de la Oposición para el Monitoreo del Proceso Electoral del Estado Trujillo.

1.4.2. Objetivos Específicos

- Estudiar la estrategia actual aplicada en el monitoreo del proceso electoral en la Sala Situacional en el Estado Trujillo.
- Identificar el soporte estadístico actual implementado para el análisis y proyección de votantes en las mesas de votación del proceso electoral en la Sala Situacional en el Estado Trujillo.
- Describir las principales necesidades tecnológicas que se requieren para mejorar el monitoreo del proceso electoral en el Estado Trujillo.
- Diseñar un sistema automatizado para la Sala Situacional de la oposición en el Estado Trujillo.
- Implementar el sistema automatizado en la Sala Situacional en el Estado Trujillo.

1.5. Antecedentes

A continuación se hace referencia a los estudios previos que se han realizado sobre las variables en estudio; Sistema Automatizado y Sala Situacional, a fin de establecer los antecedentes que guíen la investigación.

Villa y Ochoa (2013), en su estudio titulado “Sala Situacional Y Procesos Electorales En Venezuela: Importancia Del Soporte Técnico Y Estadístico”. En este trabajo se expone una reflexión sobre los lineamientos de referencia para el soporte técnico, operatividad de una Sala Situacional que funciona con el objetivo de atender un proceso electoral. De igual manera, se resumen los principales conceptos que permiten entender la Sala Situacional como una herramienta de organización del trabajo colectivo con el fin de recopilar, procesar, analizar y evaluar la información útil para sustentar y apoyar la toma de decisiones. Además se presenta un diagnóstico de la actividad en torno a una Sala Situacional electoral a partir del rol que cumplen los actores involucrados: empresas privadas consultoras, organizaciones políticas y el electorado.

Sin embargo, conviene tener presente que los resultados de un estudio electoral tienen la propiedad de reducir la incertidumbre pero no sustituye el valor del pensamiento estratégico, que hace del miembro de la Sala Situacional un ente dinámico, que supere a la típica tarea de ser organizaciones que preparan y sirven de centros de reclutamiento de votos y que estén atentos a mantener en mente que los tiempos han cambiado, las personas, en especial los más jóvenes. La tecnología más que un recurso, se muestra hoy como un tejido viscoso que alimenta la percepción sobre un mundo material y virtual, que usa como medio recursos tecnológicos de diversa índole.

Como aporte de este antecedente se presenta un soporte teórico para describir la Sala Situacional y los procesos estadísticos que se pueden emplear en las misma para resolver las dudas y problemas que desorienten los curso de las decisiones al interior de la organizaciones políticas.

Por su parte, Machado (2012), realizó una investigación titulada “Diseño del proceso para la gestión de proyectos de sistemas de información. Caso de estudio: Dirección General de Tecnología de la Información del Consejo Nacional Electoral”; la cual propone el diseño del proceso de gestión de proyectos de Sistemas de Información empleando la notación Business Process Management Notation (BPMN) y el lenguaje unificado de modelado (UML), con la finalidad que en una fase posterior se realice el desarrollo e implementación del modelo para obtener así un sistema de información basada en los compromisos de la Dirección General de Tecnología de Información (DGTI) que es la encargada de diseñar, investigar, desarrollar, implementar, mantener y gestionar los recursos en la Tecnologías de la Información.

Este proyecto aborda como problema principal la necesidad de contar con una herramienta tecnológica que pueda orientar a los líderes de proyectos en toda la gestión de proyecto, permitiéndole cumplir con la metodología del Project Management Institute (PMI), considerando diseñar el proceso con vista a la automatización cumpliendo con las fases del BPMN y la DGTI, para ello emplearon una

investigación de tipo proyectiva, ya que atiende a una necesidad detectada bajo un diseño de investigación de campo y documental.

Dicho trabajo servirá de apoyo en su estructura así como las bases teóricas propuesta para el diseño y seguimiento del proyecto que se está realizando en esta investigación.

De igual Manera, Briceño (2011), en su trabajo de investigación llamado “Estudio y Propuesta de un Sistema De Videoconferencia para Salas Situacionales de la Misión Ribas”; el presente proyecto tiene por objeto proponer un sistema de videoconferencia de escritorio que permita comunicar, en tiempo real, al personal de las Salas Situacionales de Misión Ribas, ubicadas geográficamente en lugares distantes, de una manera más fácil, cómoda, rápida y al menor costo. En la actualidad, la comunicación entre las sedes es vía telefónica o correo electrónico. Para realizar las reuniones planificadas, o no planificadas, de control y seguimiento, el personal de coordinación de cada sede se debe trasladar a la sede Nacional ubicada en Caracas. Esto implica costos recurrentes en viáticos para el traslado, alojamiento y alimentación del personal que debe trasladarse a Caracas, sin mencionar el tiempo empleado en la preparación del viaje, el estrés y la fatiga que les causan los viajes constantes. Por tal motivo, en este proyecto, se hace un estudio de la tecnología de videoconferencia, se realiza una comparación entre las diferentes soluciones de videoconferencia para escritorio y se propone la solución que mejor se adapte a las necesidades de las Salas Situacionales de la Misión Ribas.

En los aportes para esta investigación se encuentra la manera como aborda el diagnóstico de la situación actual en los procesos de comunicación y el estudio de las tecnologías disponibles. Como estructuran los objetivos, las bases teóricas y finalmente las ventajas del uso de las tecnologías para la disminución del tiempo de respuesta.

CAPÍTULO 2

Marco Teórico

En este capítulo, se hace referencia a los antecedentes o estudios desarrollados previamente sobre sala situacional, partidos políticos, monitoreo del proceso electoral, sistemas automatizados, herramientas tecnológicas, de igual manera se desarrollan las teorías que sustentan estas variables.

2.1. Sala Situacional

Villa y Ochoa (2013), define la Sala Situacional, como una herramienta de aprendizaje de vieja data y útil para adquirir competencias en la elaboración de diagnósticos y perfiles situacionales así como elementos clave en el rol de la intervención social que permita al futuro sociólogo identificar las ventajas del pensamiento estratégico en la reconstrucción de una situación y en especial, lo concerniente a las tareas recopilación, análisis, interpretación y recomendaciones en el marco de informes para trabajos en equipo.

Según Méndez y Heredia (2012), hacen mención sobre los requerimientos necesarios para la instalación de una sala situacional como requisitos indispensables los siguientes:

- **Humanos:** La Sala Situacional debe estar conformada por un equipo multidisciplinario (técnicos en estadísticas, informáticos y diseñadores gráficos comunicacionales, entre otros), responsables por la integración, unificación y análisis de los datos para producir información para la toma de decisiones gerenciales, la cual debe ser difundida a través de páginas Web, boletines, carteleras, entre otros. El equipo mínimo para iniciar el funcionamiento incluye: un transcriptor y un funcionario con conocimientos en hojas de cálculo y estadísticas básicas.
- **Equipos:** En cuanto al recurso tecnológico, es necesario contar con los equipos adecuados que faciliten el tiempo de análisis y ejecución de los procesos, entre estos: Computadoras, de alta

capacidad en cuanto a almacenamiento de datos, y memoria que permitan el uso de varias instancias o programas activos. Impresoras para la reproducción de materiales de alta calidad para carteleros (preferiblemente de color). Proyector Multimedia que permita ampliar las presentaciones en las reuniones. Cámara digital y filmadora, para identificar situaciones y recolectar evidencia de riesgos a la salud. GPS para georeferenciar en mapas y estratificar áreas de riesgo.

- **Programas o software hojas de cálculo:** Que permitan el llenado de datos o registros y para generar tablas, cuadros, gráficos, entre otros. Las utilidades fundamentales de una hoja de cálculo consisten en la facilidad de recalcular e interpretar rápidamente datos numéricos. Esta herramienta es ideal para la introducción de los datos base. En cuanto al recurso tecnológico, para conformar la sala situacional, es necesario contar con los equipos adecuados que faciliten el tiempo de análisis y ejecución de los procesos, entre estos: Computadoras, de alta capacidad en cuanto a almacenamiento de datos, y memoria que permitan el uso de varias instancias o programas activos.
- **Programas Estadísticos:** Que permitan realizar análisis de los datos tabulados; estas herramientas estadísticas tienen la capacidad de trabajar con bases de datos de gran tamaño.
- **Sistemas de información geográfica (SIG):** Programas que permiten integrar, almacenar, editar, analizar, compartir y mostrar información geográficamente referenciada; realizan consultas interactivas y son capaces de estratificar áreas de riesgo.
- **Software para el diseño y programación WEB:** Que permitan implementar la página Web de la Sala Situacional; donde se agrupe toda la información procesada en la misma.
- **Software de diseño gráfico:** Que faciliten la elaboración de imágenes, textos y figuras, con el diseño comunicacional visual para el montaje de carteras.
- **Mobiliario:** Es necesario contar al menos, con una mesa de reuniones, 6 sillas, 4 carteleros, 1 pizarra acrílica, 2 escritorios y 10 mapas de país (distribución político territorial por municipios

y/o parroquias, hidrología del país, distribución de grupos indígenas, físico- geológico, cuencas petrolíferas, regiones geográficas, relieve, suelos, puertos y aeropuertos).

- **Servicios:** Se debe contar con acceso a Internet, teléfono, fax, radio (de ser necesario por su ubicación de difícil acceso) y correo tradicional.
- **Espacio Físico o virtual:** Un espacio físico o área de cierta amplitud tanto para el montaje de indicadores, mapas, y gráficos, como para permitir reuniones colectivas multidisciplinarias; este espacio debe contar como mínimo con 20 metros cuadrados; sin embargo, puede funcionar en un espacio menor, ya que la mayor parte de la información puede estar en formato digital. Por otro lado, si se tiene un Portal electrónico o página Web, se requiere de un espacio virtual, esto significa almacenamiento en Internet a través de la página Web oficial con un enlace a la Sala Virtual o un Dominio personalizado en Internet. Esta página Web permitirá exponer en presentación digital toda la información e indicadores procesados en la Sala Situacional.
- **Recursos Financieros:** Que permitan la continuidad de las tareas y actividades de la Sala Situacional, así como la operatividad de toda la infraestructura tecnológica.

Por otro lado, Villa y Ochoa (2013) hacen referencia que en mayo de 1998, Venezuela creó el Consejo Nacional Electoral en sustitución del Consejo Supremo Electoral, dando paso a una modernización de su máxima autoridad en la materia de sufragio, y que hasta la actualidad mediante la automatización progresiva del acto del sufragio y operando bajo el ambiente de alta polarización político-ideológico ha logrado ejecutar con éxito importantes comicios: elecciones presidenciales, elecciones de gobernador, elecciones municipales, elecciones parlamentarias, referendos nacionales (revocatorio, consultivo y aprobatorio) así como otras votaciones menores como las elecciones parroquiales, renovación de la dirigencia sindical, elecciones primarias de partidos políticos, entre otros.

Es muy importante tener presente que una Sala Situacional electoral tendrá como meta elevar su capacidad de identificar y crear mecanismos que den respuesta clara y pertinente a las demandas que el electorado militante y de elector cautivo requieren en el marco de una campaña electoral. El miembro de esta Sala Situacional deberá desarrollar además habilidades y técnicas para monitorear a los medios de comunicación social que también remiten mensajes a ese elector, y oportunamente coadyuvar en reducir las brechas de información entre los entes gubernamentales y el ciudadano, entre la información que se remite y la que demanda el ciudadano.

Asimismo, debe ser garante de la idoneidad de la información o mensaje que el elector remite a todos los interesados en gobernar.

Por otra parte, se debe considerar que el elector venezolano de hoy tiene sus propias características históricas, quizás hoy día señalado de estar polarizado y que ha dado suficientes muestras de civismo con su creciente y sostenida actitud de respeto al sufragio y a la democracia como sistema político. Asimismo, cobra importancia la cualidad del elector informado, que se preocupa por opinar en los temas de interés públicos y que prefigura en sus expectativas de vida el tipo de gobernante que requiere para concretar sus demandas sociales.

2.2. Partidos Políticos

Según Sartori (2017), los partidos políticos son entidades de interés público creadas para promover la participación de la ciudadanía en la vida democrática y contribuir a la integración de la representación nacional; quienes los conforman comparten objetivos, intereses, visiones de la realidad, principios, valores y proyectos para ejecutar total o parcialmente en gobiernos de países democráticos. Estos son los encargados de presentar candidaturas a ocupar diferentes cargos políticos, para eso movilizan el llamado apoyo electoral.

2.3. Funciones De Los Partidos Políticos

La mediación entre sociedad y estado es la finalidad primordial de los partidos políticos en los regímenes democráticos enmarcados dentro de las siguientes funciones:

- La estructuración del voto.
- Reclutamiento de líderes políticos.
- Búsqueda de élites.
- Elaboración de la política pública.
- Suma o agregación de intereses.

2.4. Características De Los Partidos Políticos (Organización).

- **Dirigencia:** concentra los recursos de poder y representa el centro de la organización.
- **Candidatos:** potenciales ocupantes de los cargos públicos electivos, ya sean de carácter ejecutivo o legislativo siendo seleccionados por los demás miembros del partido.
- **Burocracia:** cuerpo administrativo.
- **Militantes:** son los que están afiliados al partido, participan activamente de manera constante.
- **Afiliados:** están inscritos en el padrón del partido y aportan a su financiación a través de cuotas periódicas, limitan su participación a la elección interna de los candidatos y autoridades.
- **Simpatizantes:** se muestran favorables a sus principios pero se mantienen apartados de la organización colaborando con sus votos y opiniones.

Índice de centralidad mide la capacidad de los presidentes y los dirigentes de imponer disciplina en sus partidos; cuando un presidente tiene un partido centralizado, es muy probable que pueda lograr imponer su programa legislativo.

2.5. Ideología.

Cada partido posee una ideología que le da claridad conceptual y lo guía en su acción política, entre estas se mencionan las siguientes:

- **Doctrinas:** conjunto de creencias que se toman como válidas.
- **Teorías:** sistematización explicativa, comprensiva e interpretativa de la realidad.
- **Plataforma:** agrupación de los principales problemas sociales, políticos y económicos.
- **Programas:** aquellos remedios para paliar los problemas.
- **Consignas:** eslóganes o lemas característicos del partido.

2.6. Poder Político

El partido político trata de ostentar el poder político, bien sea detentando la capacidad de decidir, bien sea participando en el proceso de toma de decisiones junto con otros actores políticos (partidos o no).

En los sistemas pluralistas y competitivos, existe una separación formal entre las estructuras del Estado y las estructuras del partido político. Los partidos actúan como estructuras intermedias que conectan al Estado con los ciudadanos y la sociedad civil.

2.7. Cargos Públicos Electivos

Dentro de los distintos actores políticos, los partidos tienen un estatus específico, que les separa del resto de organizaciones presentes en la sociedad civil, y les sitúa en una posición privilegiada dentro del proceso de toma de decisiones.

Ese estatus se caracteriza por ostentar el monopolio de la participación en procesos electorales plurales y competitivos, que eventualmente pueden desembocar en el nombramiento de cargos públicos electivos que, en mayor o menor medida, pueden ser controlados por los partidos o viceversa.

2.8. Monitoreo Del Proceso Electoral

Almaraz, (2010) define al monitoreo del proceso electoral como la observación y forma de recopilar información relacionada con el proceso electoral de manera sistemática, además, emite informes y evaluaciones sobre la conducción del proceso basadas en la información recabada por los observadores acreditados sin interferir en el proceso mismo. En tal sentido, el monitoreo del proceso electoral busca detectar cualquier infracción y la posibilidad de intervenir si las leyes son violadas o ignoradas en el proceso electoral, también, brinda a la autoridad correspondiente información sobre el avance del proceso electoral y las decisiones adoptadas para garantizar la integridad y neutralidad del proceso electoral.

www.bdigital.ula.ve

En lo que se refiere al proceso de monitoreo, este comprende funciones como recopilar información, formular un juicio de la información y mediación, que implica su intervención en disputas electorales con el propósito de apoyar a las partes en litigio a encontrar una solución mutuamente aceptable. Se puede inferir, que el monitoreo del proceso electoral cuenta también, con los observadores de los partidos políticos los cuales son los agentes de los partidos y los candidatos que participan en la elección; como tales, se les confiere la autoridad en la mayoría de los sistemas no sólo para monitorear el proceso electoral, sino también para intervenir si piensan que no se están respetando los requisitos legales. Asimismo, pueden ser una parte integral del proceso administrativo ya que validan papeletas y hojas de conteo mediante su firma y su participación en la administración y el escrutinio. Aunque éste es un papel voluntario en la mayoría de los sistemas, hay casos en que ni la votación ni el escrutinio se realizan sin la presencia de los observadores de los partidos políticos.

2.9. Política Contemporánea De Venezuela

Venezuela goza de un sistema de gobierno democrático desde el 23 de enero de 1958, fecha del derrocamiento de la última dictadura. A partir de ese momento, comienza un sistema de gobierno Federal, presidencialista multipartidista, caracterizado por un gran número de partidos políticos. Pero estos la mayoría de las veces terminan aliándose a las grandes cúpulas partidistas para formar coaliciones de gobierno.

Con el retorno de la democracia a Venezuela en 1958 se vuelven a legalizar los partidos políticos. AD (Acción Democrática), COPEI (Comité de Organización Política Electoral Independiente) y URD (Unión Republicana Democrática) deciden firmar el Pacto de Puntofijo, excluyendo al PCV (Partido Comunista de Venezuela) de dicho acuerdo, que consistía en la defensa de la constitución, el respeto de los resultados electorales, conformación de un gobierno de unidad nacional, entre otros; en 1960 URD se retira del Pacto luego de grandes diferencias con el presidente Rómulo Betancourt. En las elecciones de 1958 y 1963 los tres partidos dominantes se disputaron con reales posibilidades de triunfo la presidencia de la República, pero en los comicios de 1968 aparece por primera vez el fenómeno del bipartidismo en Venezuela, ya que AD y COPEI juntos obtuvieron el 57% de los votos y llegaría hasta un 93,3% en las elecciones de 1988. Durante este período el MAS fue la fuerza política más estable fuera de AD-COPEI, otros partidos como URD comenzaron a perder peso político desde entonces y algunos otros como el Movimiento Electoral del Pueblo lograron altos niveles de aceptación por breve tiempo. En las elecciones presidenciales de 1993 se rompe con el bipartidismo fundamentalmente por tres razones, al ser electo Rafael Caldera por el partido Convergencia, el crecimiento de un partido de izquierda La Causa R y la crisis social que había hecho perder credibilidad a los partidos tradicionales (AD y COPEI).

Para el año de 1998 Hugo Rafael Chávez Frías es elegido presidente y en 1999 Chávez presenta una nueva Constitución que extiende su mandato, permitiéndole buscar la reelección, cerrar el Congreso y cambiar el nombre del país a República Bolivariana de Venezuela. Por otra parte el 5 de marzo del 2013 Hugo Rafael Chávez Frías muere de cáncer, a los 58 años y el entonces vicepresidente Nicolás Maduro se convierte en mandatario interino, luego el 14 de abril del 2013 vuelven haber elecciones presidenciales ganando Nicolás Maduro convirtiéndose en presidente de la república hasta en la actualidad.

2.10. Estructura De La División Partidista En Venezuela

La transformación política de Venezuela donde los partidos tradicionales fueron desplazados en la actualidad, son el Partido Socialista Unido de Venezuela (PSUV), que es el partido de gobierno, Un Nuevo Tiempo (UNT) conformado como partido nacional en 2007 y en la actualidad es el primer partido de la coalición opositora; Primero Justicia (PJ), Podemos y el PPT (Patria Para Todos). Por otro lado, desde enero 2019, se visualiza la disputa del poder encabezada por dos hombres que han reclamado la presidencia: uno de ellos es Nicolás Maduro, quien ha controlado el poder durante seis años, y el otro es Juan Guaidó, líder de la oposición y presidente del Parlamento. En este sentido, tras asumir la presidencia interina, Guaidó propuso tres objetivos centrales para su estrategia política: el cese de la usurpación del gobierno de Nicolás Maduro, el establecimiento de un gobierno de transición impulsado por la Asamblea Nacional y la celebración de elecciones libres y transparentes.

Al respecto señala López (2019), la situación política de Venezuela para 2019 se enmarca en un contexto abiertamente autoritario y altamente conflictivo, donde la sociedad ya da señales de que participará en una confrontación de resultado incierto. Al igual que otros años los conflictos ocurridos en el pasado 2019, encuentra a las fuerzas que apoyan al presidente Maduro, poderoso pero minoritario,

aferrándose al poder, mientras que los actores nacionales opositores se aprestan de nuevo a buscar el anhelado cambio democrático global y estructural de la crisis en Venezuela.

2.11. SMS (Short Message Service)

Larrea, (2012) define SMS (short message service) como “un sistema que ofrecen los teléfonos, principalmente móviles, para enviar y recibir mensajes escritos de una longitud determinada (160 caracteres)”. En este sentido podemos decir que el sistema SMS, junto con el chat, móvil o Internet no son solamente productos de las nuevas tecnologías, sino que son sistemas que condicionan las situaciones de comunicación.

Así mismo, Calderón y Vicuña (2012) expresan que los SMS significa Short Message Service, lograron la masificación de su uso desde hace pocos años de la mano de los teléfonos celulares, pese a que existen desde 1985 cuando un grupo de técnicos franceses y alemanes desarrollaron y establecieron un estándar para el servicio de envío de mensajes cortos que hoy utiliza la sociedad en todo el mundo, es una forma rápida y más usada de hacer saber algún comunicado a otro usuario de un teléfono celular. Por otra parte, los mensajes de texto son procesados por un «centro de servicio de mensajes cortos» (Short Message Service Center, SMSC) que se encarga de almacenarlos hasta que son enviados y de conectar con el resto de elementos de la red GSM (del inglés Global System for Mobile communications, traducido Sistema Global para las Comunicaciones móviles). Además, cuando un usuario envía un SMS, o lo recibe, se incluyen con su payload (carga útil) al menos los siguientes parámetros:

- Marca de tiempo de envío (timestamp).
- Periodo de validez del mensaje, desde una hora hasta una semana.
- Número de teléfono del remitente y del destinatario.
- Número del SMSC que ha originado el mensaje.

Con este modo se asegura el correcto procesamiento del mensaje en el SMSC y a lo largo de toda la cadena.

Ahora bien, los avances tecnológicos de los teléfonos móviles y las redes de comunicación no han conseguido eliminar por completo la presencia o la utilidad del SMS, ya que todavía al día de hoy la adopción de los smartphones (Teléfonos Inteligentes) no es absoluta. Por otro lado, el miedo a los ataques tales como el robo de identidad hace que muchas personas eviten sistemas como WhatsApp y continúen usando SMS para sentirse más seguras. Además, sigue siendo un medio más privado, ya que no da al receptor más información que el número de teléfono del remitente, y no le permite saber si se encuentra conectado, del mismo modo que este último no puede saber si el destinatario ha leído sus mensajes. Sin embargo, el SMS es perfecto para comunicar planes, ideas, o envío de información que no requieren una llamada telefónica el mismo se ha convertido en la forma de comunicación de preferencia para muchos usuarios, tanto en su vida personal como profesional, puede ampliar las aplicaciones web y mejorar la experiencia informática en general con la comunicación móvil.

2.12. Python

Molina, Loja y otros (2016) define Python como “un lenguaje de programación interpretado e interactivo, capaz de ejecutarse en una gran cantidad de plataformas. Se desarrolla como un proyecto de código abierto, administrado por PYTHON Software Foundation.” Así mismo, Obando y Ramírez (2016) Python es un lenguaje de programación de propósito general aplicado a menudo en roles de scripting (secuencia de comandos). Comúnmente se define como un lenguaje de scripts orientado a objetos una definición que combina soporte para POO (Programación Orientada a Objeto) con una orientación general hacia funciones de script, destacando como las principales características citadas por usuarios de Python son las siguientes:

- **Calidad del software.** Para muchos, Python se centra en la legibilidad, la coherencia y la calidad del software en general se distingue de otras herramientas de scripting en el mundo. El código de Python está diseñado para ser legible, por lo tanto reusable y mejorable, mucho más que el de lenguajes de script tradicionales.
- **Productividad del desarrollador.** Python aumenta la productividad muchas veces más allá que lenguajes compilados o estáticamente escritos como C, C++ y Java. El código de Python es habitualmente entre un tercio y un quinto del tamaño de su equivalente en código C++ o Java. Esto significa que hay menos que escribir, depurar, y menos para mantener después de hecho.

También, Python puede ser utilizado en diversas plataformas y sistemas operativos, entre los que podemos destacar los más populares, como Windows, Mac OS X y Linux.

Pero, además, puede funcionar en smartphones al ser un lenguaje multiparadigma brinda innumerables beneficios al permitir al usuario trabajar bajo varios estilos: programación orientada a objetos, programación funcional, entre otros.

Otro aspecto importante a considerar es que permite la facilidad de extensión, esto quiere decir que se puede escribir nuevos módulos de manera fácil en bajo lenguaje como C o C++ y se puede incluir para aplicaciones que necesiten una interfaz programable.

2.13. SL4A

Desde la perspectiva de López (2012), es una herramienta que permite desarrollar, ejecutar scripts e interactuar con intérpretes directamente en el propio dispositivo Android. SL4A incluye distintos lenguajes de scripting que tienen acceso a las APIs (interfaces de programación de aplicaciones) de Android mediante llamadas remotas (RPCs) a un servidor implementado como una aplicación Java estándar de Android. Además, la capa de secuencias de comandos para Android (SL4A) es una biblioteca que proporciona una plataforma para escribir de manera simple scripts para automatizar

tareas en un dispositivo Android usando idiomas distintos de JAVA. SL4A, se anunció por primera vez en el Blog de Google Open Source en junio de 2009; el proyecto vio la luz del día principalmente gracias a los esfuerzos de Damon Kohler y ha crecido gracias a las contribuciones de otros desarrolladores.

En este mismo orden de ideas, SL4A es la abreviatura de "Scripting Layer for Android", esto es, una capa de software para utilizar scripts en Android. Es una librería que permite la creación y ejecución de scripts, escritos en distintos lenguajes, en dispositivos Android.

Es importante destacar que, estos scripts tienen acceso a muchas de las APIs disponibles en los programas Android desarrolladas normalmente en Java, pero con una interfaz simplificada, los scripts pueden ejecutarse en un terminal, o como servicio utilizando la arquitectura de servicios en Android.

2.14. Aplicación Qpython

Para Peñaloza (2016), QPython es un motor de script que ejecuta programas de Python en dispositivos Android. También puede ayudar a los desarrolladores a desarrollar aplicaciones de Android. Así mismo, QPython, es una opción muy viable para la programación Android desde el mismo terminal como herramienta muy poderosa, la oportunidad de programar en Python desde nuestro teléfono móvil Android, y más importante que eso, poder acceder a los sensores del propio móvil entre otras cosas.

En este mismo sentido, QPython es un motor de secuencias de comandos que se ejecutan en los dispositivos Android, permite correr proyectos y scripts python en nuestros dispositivos Android, contiene un intérprete Python, consola, un editor, y la librería SL4A para Android, ofrece un kit que permite desarrollar fácilmente proyectos y scripts de Python en los dispositivos Android.

Por otra parte, para realizar este proyecto, se necesita algún mecanismo que permita realizar una aplicación Android en código Python, a través de la búsqueda de la herramienta que admitiese esta acción, así como su estudio y posterior aplicación, eran, en inicio, la parte más importante a realizar, encontrar la posibilidad de ejecutar código Python, sobre un dispositivo Android es algo que se ha logrado mediante la herramienta Qpython. Dicha herramienta es bastante actual y novedosa, características siempre positivas para encontrar información en Internet, pero también estas características conllevan unas desventajas como que la documentación aún no esté muy bien estructurada.

2.15. Modelo Cliente-Servidor

En lo que respecta al Modelo Cliente-Servidor, IBM (2012) lo define como la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas, el modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o clientes, resultan en un trabajo realizado por otros computadores llamados servidores.

Por otro lado, “La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.” (Sepulveda, 2009 citado por Murillo y Caamaño 2010). Según la definición de la referencia anterior, es un modelo en el que el procesamiento requerido para realizar una tarea se divide en dos o más procesos cooperantes. Se denomina cliente al proceso que solicita los recursos, y servidor al proceso que responde a las

solicitudes. Para que pueda existir comunicación entre el cliente y el servidor debe existir la infraestructura necesaria para los mecanismos de direccionamiento y transporte.

2.16. XML-RPC

Para Gómez XML-RPC (2016) lo define como una especificación y un conjunto de implementaciones que permiten que el software se ejecute en sistemas operativos dispares, que se ejecutan en diferentes entornos para realizar llamadas de procedimiento a través de Internet, es un procedimiento remoto que llama usando HTTP como transporte y XML como codificación. XML-RPC está diseñado para ser lo más simple posible, al tiempo que permite transmitir, procesar y devolver estructuras de datos complejas. Es una especificación y un conjunto de implementaciones que permiten que software corra en sistemas operativos diferentes, corriendo en ambientes distintos, teniendo en cuenta que una de las ventajas de utilizar XML-RPC es que la comunicación se puede realizar en sistemas operativos diferentes y en entornos diferentes como lo son PERL, LUA, PHP, JAVA entre otros. Los lenguajes de programación mencionados poseen librerías XML-RPC para hacer más flexible la programación de aplicaciones que utilicen llamadas a procedimientos remotos. El entorno o lenguaje de programación utilizado por el cliente y el servidor pueden ser diferentes.

Para Fernández (2014) Un sistema de procedimientos remotos, RPC por sus siglas en inglés (Remote Procedure Call) es una comunicación entre procesos que permite a un programa de computadora ejecutar una subrutina o proceso en un espacio de memoria diferenciado (usualmente en otro equipo, aunque esto no es un requisito indispensable) sin que el diseñador de esta sub-rutina tenga que preocuparse explícitamente de los detalles de esta interacción remota, abstrayéndose de ella con el fin de conseguir que el diseñador se despreocupe de las comunicaciones entre ambos equipos, se diseña un protocolo (llamado también RPC), que permite al usuario encapsular las comunicaciones dentro de las RPC, suponiendo un gran avance frente a los sockets usados previamente y que requerían una gran

atención por parte del programador. Hoy en día la tendencia es a utilizar XML como lenguaje de descripción del interfaz (define de qué manera se muestra la información que define los procedimientos y los datos que estos pueden necesitar o devolver) y HTTP como protocolo de red, dando lugar a lo que se han venido en denominar servicios web, como ejemplos de estos cabe citar SOAP o XML-RPC.

Según Primo (2012) el Protocolo de Transferencia de HiperTexto (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. La especificación completa del protocolo HTTP 1/0 está recogida en el RFC 1945. Fue propuesto por Tim Berners-Lee, atendiendo a las necesidades de un sistema global de distribución de información como el World Wide Web. Desde el punto de vista de las comunicaciones, está soportado sobre los servicios de conexión TCP/IP, y funciona de la misma forma que el resto de los servicios comunes de los entornos UNIX: un proceso servidor escucha en un puerto de comunicaciones TCP (por defecto, el 80), y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

Veamos a continuación como funciona, de forma general, un sistema de procedimientos remotos:

- El cliente inicia el proceso RPC enviando una petición a un servidor conocido, en la petición le debe indicar al servidor qué procedimiento debe ejecutar y suministrarle los parámetros necesarios (si se requieren).
- El servidor envía una respuesta al cliente y la aplicación continúa su ejecución. En este punto hay múltiples variantes dependiendo del sistema concreto de RPC que se utilice. Por norma general el cliente queda en un estado de bloqueo (esperando a que termine la tarea en el servidor) hasta que este le envíe la respuesta o bien se cumpla un tiempo máximo de espera (timeout).

Se comprende entonces, como un protocolo muy simple ya que solo define unos cuantos tipos de datos y comandos útiles, además de una descripción completa de corta extensión. La simplicidad del XML-RPC está en contraste con la mayoría de protocolos RPC que tiene una documentación extensa y requiere considerable soporte de software para su uso.

2.17. Servidor Web

De acuerdo con Cases (2014), un servidor es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets, banners, etc). Estos servidores web utilizan el protocolo http. Expresa Cases que los servidores web están alojados en un computador que cuenta con conexión a Internet. El Servidor Web, se encuentra a la espera de que algún navegador le haga alguna petición, como, por ejemplo, acceder a una página web y responde a la petición, enviando código HTML mediante una transferencia de datos en red. Se deduce entonces, que el rol principal de un servidor web es almacenar y transmitir el contenido solicitado de un sitio web al navegador del usuario. Para cumplir con sus funciones el servidor deberá tener la capacidad de estar siempre encendido para evitar interrumpir el servicio que les ofrece a sus clientes. Si dicho servidor falla o se apaga, los usuarios tendrán problemas al ingresar al servicio que proporciona el servidor.

2.18. Apache

Para González (2014) define Apache como “El proyecto de servidor HTTP Apache”, el cual fue desarrollado como código fuente-abierto (open-source) para sistemas operativos modernos incluyendo UNIX y Windows®. El objetivo de este proyecto es el de proveer un servidor seguro, eficiente y extensible que a la vez provea de servicios HTTP que estén dentro de los lineamientos del estándar con el mismo nombre. Ha sido el servidor web más popular en el internet desde hace ya 17 años. (The Apache Software Foundation, 2012).

Es por ello que, Apache es un servidor web de código libre robusto cuya implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo que, usando Internet y la web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada. Estos voluntarios se conocen como el Apache Group. Además del Apache Group, cientos de personas han contribuido al proyecto con código, ideas y documentación.

Es así como, el servidor web Apache se ha convertido en el servidor web más utilizado en el mundo debido a sus altas prestaciones y desempeño, además de ser gratuito, lo cual contribuye a su rápida expansión y posicionamiento. La configuración de este servidor web para aquellas personas que posean un conocimiento medio del sistema operativo Linux no debe ser un problema, pero resulta en ocasiones complicado e intimidante enfrentarse a los archivos de configuración del servidor sin una guía o con la base de la información fragmentada y de lenguaje oscuro que se puede obtener en la web.

www.bdigital.ula.ve

2.19. Php

Considerando el criterio de Mateu (2004) PHP, cuyas siglas responden a un acrónimo recursivo (PHP: hypertext preprocessor), es un lenguaje sencillo, de sintaxis cómoda y similar a la de otros lenguajes como Perl, C y C++. Es rápido, interpretado, orientado a objetos y multiplataforma. Para él se encuentra disponible una multitud de librerías. PHP es un lenguaje ideal tanto para aprender a desarrollar aplicaciones web como para desarrollar aplicaciones web complejas. PHP añade a todo eso la ventaja de que el intérprete de PHP, los diversos módulos y gran cantidad de librerías desarrolladas para PHP son de código libre, con lo que el programador de PHP dispone de un impresionante arsenal de herramientas libres para desarrollar aplicaciones.

Por ende, PHP suele ser utilizado conjuntamente con Perl, Apache, MySQL o PostgreSQL en sistemas Linux, formando una combinación barata (todos los componentes son de código libre), potente y versátil. Tal ha sido la expansión de esta combinación que incluso ha merecido conocerse con un nombre propio LAMP (formado por las iniciales de los diversos productos). Apache, así como algunos otros servidores web, Roxen entre ellos, puede incorporar PHP como un módulo propio del servidor, lo cual permite que las aplicaciones escritas en PHP resulten mucho más rápidas que las aplicaciones CGI habituales, así que, el lenguaje puede ejecutarse en prácticamente todos los sistemas operativos actuales y en múltiples servidores web. Este también soporta una amplia variedad de bases de datos y cuenta con múltiples librerías para ejecutar procesos comunes, lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales.

2.20. PostgreSQL

Según Kasián y Reyes (2012) PostgreSQL es un potente motor de bases de datos relacionales reconocido por su fiabilidad, integridad de datos y correcto desempeño, como así también por su alta portabilidad a los principales sistemas operativos Linux, Unix (sus derivados) y Windows. Su código fuente está disponible bajo licencia de código abierto por lo que es posible su uso, modificación, distribución y documentación del Servidor de Base de Datos.

De igual manera, Zea, Molina y otros (2017) PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado (Sobre PostgreSQL). Este sistema utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (Sobre PostgreSQL). También se le conoce como un sistema de base de datos relacionales

es un sistema que permite la manipulación de acuerdo con las reglas del álgebra relacional. Los datos se almacenan en tablas de columnas y renglones. Con el uso de llaves, esas tablas se pueden relacionar unas con otras.

2.21. Modelo Vista Controlador (MVC)

Para Espitia, Amaro y otro (2016) El Model-View-Controller o Modelo Vista Controlador (MVC) es un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista, o a cualquier parte del sistema puedan ser hechas con un mínimo impacto en el componente del modelo de datos o en los otros componentes del sistema. Este patrón cumple perfectamente el cometido de modular un sistema, siendo su propósito principal el de simplificar la implementación de aplicaciones de acuerdo a las peticiones de los usuarios y los datos a desplegar.

Por otra parte, MacWilliams citado por Sagredo, Gamaliel y otro (2012), señala que el patrón Modelo, Vista y Controlador (MVC) es el más extendido para el desarrollo de aplicaciones donde se deben manejar interfaces de usuarios, éste se centra en la separación de los datos o modelo, y la vista, mientras que el controlador es el encargado de relacionar a estos dos.

- **Modelo:** Es la representación de la información que maneja la aplicación. El modelo en sí son los datos puros que puestos en contexto del sistema proveen de información al usuario o a la aplicación misma.
- **Vista:** Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación Web, la “Vista” es una página HTML con contenido dinámico sobre el cual el usuario puede realizar operaciones.
- **Controlador:** Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando el Modelo en caso de ser necesario.

CAPÍTULO 3

Marco Metodológico

En cuanto a la elaboración de la aplicación web se utilizó el modelo desarrollo iterativo e incremental el cual es definido por Victor Basili & Joe Turner (2019), como un sistema de software incremental, permitiendo al desarrollador aprovechar lo que se va aprendiendo durante el desarrollo de las versiones tempranas (versiones beta o versiones iniciales), incrementales y entregables del sistema. Por lo general, el aprendizaje viene tanto del desarrollo como del uso del sistema, donde sea posible. Por otro lado, los pasos claves en el proceso consisten en empezar con una implementación sencilla de un subconjunto de los requisitos del software y mejorar iterativamente la evolución secuencial de versiones hasta que el sistema está implementado y en cada iteración, se hacen modificaciones en el diseño a la misma vez que añadimos nuevas funcionalidades. Además, el modelo incremental de gestión de proyectos tiene como objetivo un crecimiento progresivo de la funcionalidad. Es decir, el producto va evolucionando con cada una de las entregas previstas hasta que se amolda a lo requerido por el cliente o destinatario.

A tal efecto, este enfoque que se usó inicialmente para proyectos de software aunque más tarde se aplicó a otros sectores, establece entregas parciales mediante un calendario de plazos. En cada una de ellas, el producto debe mostrar una evolución con respecto a la fecha anterior; nunca puede ser igual. Vemos entonces, que una de las claves para que esto se haga efectivo es la evaluación de las etapas y los responsables del proyecto deben analizar si los resultados parciales son los esperados y si, sobre todo, apuntan al objetivo principal. De no ser así, deberán intervenir en él e implementar las soluciones que la situación requiera.

Considerando que el sistema puede dividirse en etapas, estableciendo la toma, envío, procesamiento y despliegue de estadística de la data como se muestra en la figura 2, se determinó emplear el Desarrollo

iterativo e incremental, en virtud que combina los elementos del modelo en cascada con la filosofía de construcción de prototipos.

En base a lo expuesto por Alvear, J. y Mazón, J. (2007) define el modelo cascada como el primer modelo de proceso de desarrollo de software que se publicó y que se derivó de otros procesos de ingeniería. Las principales etapas de este modelo se transforman en actividades fundamentales de desarrollo: 1) Análisis y definición de requerimientos. 2) Diseño de sistemas y de software. 3) Implementación y prueba de unidades 4) Integración y prueba del sistema 5) Operación y mantenimiento.

Por su parte, Garduño (2015) señala que los prototipos son los encargados de iniciar el proceso con la comunicación que tiene el cliente con el desarrollador, en donde se identifican los requisitos y objetivos globales proporcionando una vista general del sistema, con ello se permite que todo el sistema o algunas de sus partes se construyan rápidamente para comprender, aclarar o recabar nuevos requisitos, de esta forma el usuario y el programador revisan la calidad del software, lo que permite una mayor facilidad de mantenimiento en cada prototipo parcial o total. Por lo tanto, la construcción por prototipos se resume en las siguientes actividades: Identificación de requerimientos son las necesidades básicas de información del usuario, entradas y salidas del sistema, desarrollo de un prototipo, el diseñador crea un prototipo funcional, uso del prototipo: el usuario prueba el sistema para determinar si cumple con sus necesidades, haciendo sugerencias u observaciones sobre el manejo del mismo, mejora del prototipo: el desarrollador toma nota de todas las observaciones o mejoras que solicita el usuario, depura el prototipo y la penúltima y última etapa se repiten hasta que el usuario quede satisfecho.

Buscando la construcción en forma secuencial de cada una de las etapas planteadas en la figura 2.

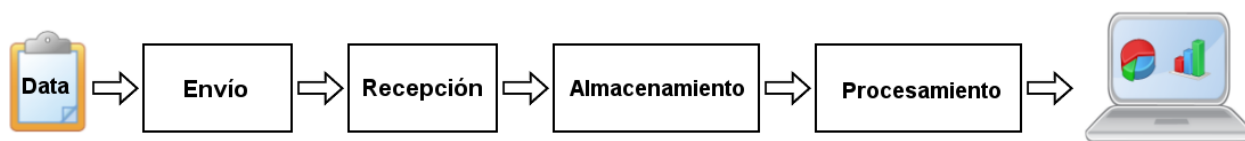


Figura 2. Etapas observadas en la transmisión/recepción y procesamiento de data de un evento electoral.

Otra razón que se considera este modelo, es que es útil cuando no se cuenta con una dotación de personal suficiente, y justamente la división del problema en etapas (iteraciones) es quien permite trabajar, para ir logrando objetivos específicos. Es importante resaltar que las iteraciones no son independientes, la salida de cada bloque es la entrada del siguiente en cada una de las etapas, esto garantiza un avance en cada iteración. Por otro lado, OBS Busines School (2016), presenta las diferentes etapas donde el modelo iterativo e incremental no es riguroso, lo que permite ajustar las fases de acuerdo al proyecto, en este caso particular se establecieron las siguientes fases:

3.1. Fase 1.- Requerimientos: Son los objetivos centrales y específicos que persigue el proyecto.

3.2. Fase 2.- Definición de las tareas y las iteraciones: Teniendo en cuenta lo que se busca, el siguiente paso es hacer una lista de tareas y agruparlas en iteraciones que tendrá el proyecto. Esta agrupación no puede ser aleatoria. Cada una debe perseguir objetivos específicos que la definan como tal.

3.3. Fase 3.- Diseño de los incrementos: establecidas las iteraciones, es preciso definir cuál será la evolución del producto en cada una de ellas. Cada iteración debe superar a la que ha precedido. Esto es lo que se denomina incremento.

3.4. Fase 4.- Desarrollo del incremento: posteriormente se realizan las tareas previstas y se desarrollan los incrementos establecidos en la etapa anterior.

3.5. Fase 5.- Validación de los incrementos: al término de cada iteración, se debe verificar las salidas de los incrementos a través de un conjunto de entradas (test). Si no da los resultados esperados se debe volver a la iteración anterior y buscar la causa de ello. Si los resultados son los esperados, se procede a documentarlos y se pasa a la siguiente fase. La documentación de cada incremento permite realizar mínimos esfuerzos a la hora de entregar el producto.

3.6. Fase 6.- Integración de los incrementos: Una vez que se valide el incremento, dando forma a lo que se denomina línea incremental o evolución del proyecto en su conjunto, este incremento ha contribuido al resultado final.

3.7. Fase 7.- Entrega del producto: Cuando el producto en su conjunto ha sido validado y se confirma su correspondencia con los objetivos iniciales, se procede a su entrega final.

www.bdigital.ula.ve

CAPÍTULO 4

Diseño Del Sistema

En este capítulo se habla de la solución tecnológica aplicada para resolver la problemática, la cual es desarrollar un sistema web para la consulta y el envío de reportes desde los centros electorales, a través de mensajes de texto SMS, siendo el objetivo primordial brindar la facilidad de comunicación entre el personal que está desplegado a nivel estatal con la sala situacional. Así pues, el diseño de la aplicación se realizó teniendo en cuenta los requerimientos y la accesibilidad que tienen los usuarios con las tecnologías; en esta etapa se relacionaron los requisitos, arquitectura del sistema, la descripción de las vistas arquitectónicas, el diseño de la interfaz de usuario/sistema, y el diseño de la base de datos.

Por otro lado, el valor de la factibilidad operativa radica en el estudio de la situación actual del sistema, en la cual se puede determinar los cambios en el proceso, para que la sala situacional una vez implementado la aplicación web pueda mejorar su rendimiento. Actualmente en la sala situacional no existe un medio digital automatizado en la cual se pueda llevar la información, por tal razón la persona encargada de realizar los registros lo hace de forma manual. De esta manera, la aplicación fue diseñada para poder almacenar la información fiable y actualizada de los reportes enviados desde los centros de votación, además la aplicación web no presenta ningún grado de dificultad, ya que es operable desde un teléfono móvil inteligente o una computadora, con una interfaz amigable, permitiendo facilidad de manejo. También, se considera como viable operativamente porque además de contar con personal altamente calificado que brindará soporte en la parte tecnológica, se realizará capacitaciones a los diferentes usuarios del sistema puesto en práctica, que será de gran ayuda el día del evento electoral en la sala situacional.

4.1. Descripción de actores

Los actores son aquellos que interactúan con el sistema, representan a todos los que necesitan intercambiar información con el sistema, cabe señalar que el modelado de actores implica una jerarquía la cual es representada en una escala ordenada según un criterio de mayor a menor importancia o relevancia dentro del sistema.

Tabla 1. Actores que cooperan en los centros de votación.

ACTOR	DESCRIPCIÓN	PERFIL
SÚPER-MONITOR.	Es el que redacta los reportes suministrados por los monitores, está en constante comunicación con la sala situacional, envía la data a través de un mensaje de texto (sms).	El súper-monitor es el encargado de coordinar el personal, la máxima autoridad en cada centro, es una persona capacitada que conoce el proceso.
MONITORES.	Son los que recolectan la información una vez iniciado el proceso de votación, cada cierto periodo de tiempo reportan al súper-monitor los datos obtenidos de cada mesa de votación.	Los Monitores están a la orden de sus coordinadores (súper-monitor), se encuentran ubicados por cada una de las mesas contabilizando el número de votantes y votos a favor.

Tabla 2. Actores que laboran en la sala situacional.

ACTOR	DESCRIPCIÓN	PERFIL
ADMINISTRADOR.	Tiene acceso total a todo el sistema, puede consultar, insertar, eliminar, modificar, actualizar cualquier usuario o reporte de manera manual en caso de equivocaciones.	El administrador es el responsable de toda la sala situacional, la máxima autoridad, altamente capacitado, conoce todo los procesos del sistema.
SUPERVISOR-ESTATAL.	Puede hacer consultas a todos los niveles: reportes generales, municipales, parroquiales, centros específicos.	El usuario estatal es el siguiente en la jerarquía, puede ser cualquier persona de confianza que conozca el uso del sistema y pueda hacer uso de este.
SUPERVISOR-MUNICIPAL.	Pueden consultar exclusivamente el municipio asignado, las parroquias y sus respectivos centros	Los perfiles municipales son cualquier persona que esté autorizada y que tenga la capacidad de ingresar al sitio web.

4.2. Requerimientos

En lo que se refiere, a los requerimientos son la pieza fundamental de un proyecto de desarrollo de software, describiendo técnicamente las funciones, el desempeño y las características que el producto

debe tener. Cabe destacar que, el presente proyecto propone un sistema que sea capaz de permitir crear un sistema automatizado para la Sala Situacional de la Mesa de la Unidad Democrática o Partidos Políticos de la Oposición para el Monitoreo del Proceso Electoral del Estado Trujillo. Envío, recepción, almacenamiento, procesamiento de la data mediante el uso de herramientas tecnológicas.

4.2.1. Requerimientos de Entorno

De igual manera los requerimientos de entorno, son aquellos que rodean al sistema, lo usan y lo necesitan como una fuente de los servicios necesarios para que funcione. El sistema deberá ser implementado en un servidor de aplicaciones web y un teléfono inteligente disponible las 24 horas del día que ocurra el evento electoral, con los siguientes requisitos:

- Teléfono que pueda enviar mensaje de texto (SMS).
- Teléfono Inteligente.
- Sistema Operativo Versión Android 5.1 o superior.
- Aplicación QPython3 Instalada Versión 1.3.2 o inferior.
- Computador Requerimientos Mínimos Pentium 4, Disco Duro 80gb, Memoria Ram 512 mb.
- Sistema Operativo Debian 9 Stretch o superior.
- Apache 2, Php, PostgreSQL.
- Composer y Curl.
- XML para Php.
- Librería XML-RPC para Php.

4.2.2. Requerimientos Funcionales

En cuanto a los requerimientos funcionales, son aquellos que describen lo que el sistema debe de hacer, es decir, son funciones que debe ser capaz de realizar el sistema. En el presente sistema web se determinaron los siguientes requisitos:

- Gestión de roles.
- Gestión administrativa del proceso de recolección de datos.
- Gestión de recolección de datos.
- Generación de estadísticas relacionadas al sistema web.
- Consultas y recuperación de la información.

4.2.3. Requerimientos no Funcionales

Por otra parte, los requerimientos no funcionales son aquellas restricciones físicas del sistema, estos surgen de la necesidad del usuario, en otras palabras, son las restricciones de la aplicación, los atributos de calidad, los límites de memoria, requerimientos de seguridad, restricciones de software, restricciones de hardware, entre otros.

En el caso de este sistema se encontraron los siguientes requisitos:

- La arquitectura de software del teléfono inteligente debe cumplir con los siguientes requisitos: sistema operativo Android donde debe estar activa la aplicación Qpython3 ejecutando un script (código).
- La arquitectura de software del servidor debe contar es siguiente características: Sistema operativo libre donde debe estar corriendo el servicio Apache2 Http Server como servidor web, PostgreSQL como servidor de base de datos, Frameworks BOOPSTRAP, CSS, CSS3, HTML, como herramientas de desarrollo para que sea responsivo y así, ser visualizado en cualquier

dispositivo, el lenguaje de programación debe ser PHP para el código ejecutado del lado del servidor.

- El acceso al sistema web puede realizarse a través de cualquier conexión estándar de internet.
- Todas las consultas al servidor de base de datos deben ser rápidas y eficientes, de modo que cualquier eventualidad pase desapercibida para los usuarios.
- La interfaz del sistema debe ser intuitiva para facilitar la navegación de los usuarios.

4.3. Desarrollo de la Aplicación

Después de realizar el análisis y establecer el diseño del sistema se llegó a constituir cómo será el Aplicación Web, de acuerdo a los requerimientos establecidos por todos los participantes tanto internos como externos, teniendo siempre presente que la tecnología disponible no es siempre la misma, sino que es muy dinámica, está sometido a continuas novedades que aparecen a diario; por este motivo que es altamente aconsejable, antes de proceder al diseño de la arquitectura del sistema, identificar los elementos con los que se dispone:

4.4. Plataforma De Comunicaciones

Los Centros cuenta con una plataforma de comunicación a través de uno o varios teléfonos celulares, luego de recolectada la data de sus respectivas mesas se procede a enviar un mensaje de texto (SMS) con el siguiente formato:

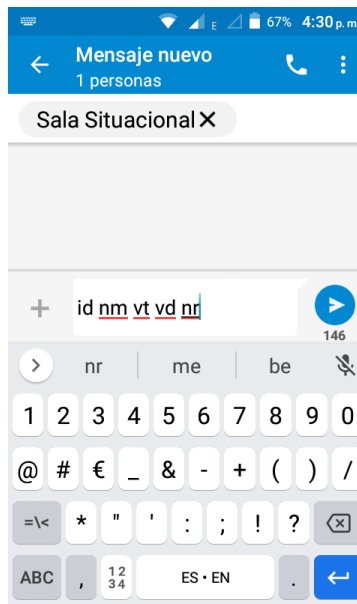


Figura 3. Captura pantalla etapa envió SMS.

Descripción del cuerpo del mensaje figura3:

- **id** = Número de identificación.
- **nm** = Número de Mesa.
- **vt** = Cantidad de Votantes.
- **vd** = Cantidad de Votantes a Favor
- **nr** = Número de Reporte.

En la siguiente imagen se ilustra un ejemplo aleatorio de un reporte:

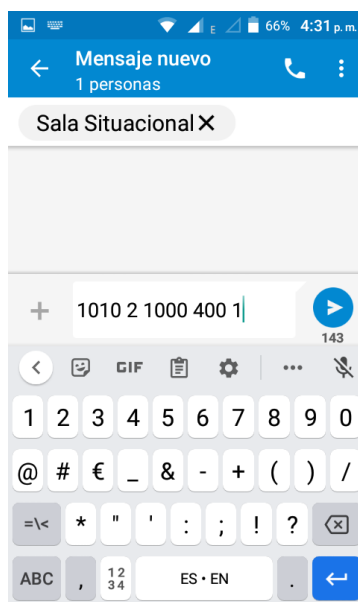


Figura 4. Captura pantalla ejemplo envió SMS.

Descripción del cuerpo del mensaje figura4:

- **1010** = Número de identificación.
- **2** = Número de Mesa.
- **1000** = Cantidad de Votantes.
- **400** = Cantidad de Votantes a Favor.
- **1** = Número de Reporte.

Luego se enviar el reporte a la sala situacional, los cuales serán utilizados para dar marcha a la funcionalidad de la solución tecnológica a implantar y además facilitar las comunicaciones, entre el personal del área usuaria, los cuales se encontrarán monitoreados por los supervisores correspondientes quienes tendrán acceso a dicha información en cualquier momento dentro de la sala situacional.

4.5. Plataforma de hardware

4.5.1. Hardware Aplicación Móvil

La Sala Situacional cuenta con un tipo de plataforma escalable, en base a las necesidades usuarias de manejo de aplicaciones de sistemas, siendo el componente principal del sistema una codificación ejecutada en una aplicación móvil la cual necesita ser instalada en dispositivos móviles con los siguientes requisitos mínimos:

- Un dispositivo móvil (tableta que pueda recibir SMS o Smart phone).
- Sistema operativo Android versión 5.1 o mayor.
- Capacidad mínima en memoria RAM de 1GB.

Adicional a estos requerimientos es de gran importancia poseer una conexión inalámbrica o una conexión de datos veloz y estable.

Nota: La conexión a internet es necesaria si el dispositivo móvil esta fuera de la red de la sala situacional, en caso de no tener acceso a internet se puede crear una red local que se conecte el dispositivo móvil con el servidor.

4.5.2. Hardware Del Servidor

Para la realización y ejecución se hace necesario la instalación de una aplicación web la cual constituye la parte administrativa del sistema, en un servidor web con los siguientes requisitos mínimos:

- Cualquier procesador con una velocidad mínima de 2 GHz.
- Una capacidad mínima en memoria RAM de 512mb.
- Se requiere un espacio mínimo en disco duro de 10 GB.

4.6. Software de Desarrollo

4.6.1. Dispositivo Móvil (Smart phone)

En esta fase, con la información previamente obtenida se empieza a realizar los algoritmos necesarios para lograr cumplir con los requerimientos que necesita el usuario, luego de llegar los SMS se almacenan en el buzón del teléfono inteligente, se observa el esquema de despliegue del sistema en el cual el cliente envía una petición al servidor web/proxy apache2.

La arquitectura de la solución está orientada a servicios, el proyecto consta de la Aplicación Móvil en ejecución:

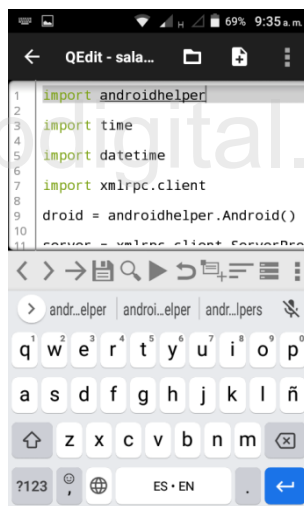


Figura 5. Aplicación Activa con su respectivo código a ejecutar.

Esta etapa se emplea el código en una forma que sea de fácil comprensión por la Aplicación Móvil, donde describimos los puntos importantes del código fuente que establece las conexiones cliente-servidor.

```

1 import androidhelper
2
3 import time
4
5 import datetime
6
7 import xmlrpc.client

```

Figura 6. Librerías Python.

En la imagen se importan las librerías para las iteraciones del cliente en lenguaje de programación Python, en estos fragmentos de código la línea número 7 es la más importante, ya que llama la librería del protocolo de comunicaciones XML-RPC en este caso la parte del cliente.

```

11 server = xmlrpc.client.ServerProxy("http://mud-sala.sytes.net:11180/xmlrpc/pruebaspy/enviando/servidor/recibiendo10.php")

```

Figura 7. Envía la petición de comunicación con el servidor.

En la línea 11 se realiza la llamada al servidor, establece la comunicación del server, puerto o protocolo, en este caso la URL que cuenta con un sub-dominio <http://mud-sala.sytes.net:11180> apuntando al código fuente del servidor (**recibiendo10.php**).

```

13 while True:
14     SMSMsgs = droid.smsGetMessages(True, 'inbox').result

```

Figura 8. Recorrido del buzón de mensaje de texto.

Comienza el bucle con una estructura repetitiva (**while**) mientras esta sea verdadera comienza el ciclo, en la línea 14 se hace un recorrido de todo el buzón de mensajes de texto almacenados en el dispositivo, donde **smsGetMessages** devuelve una lista de todos los mensajes.

```
23
24     print(server.mail.send(message['address'], '', message['body'], mydate.strftime('%Y-%m-%d %H:%M:%S'), '', '', ''))
25
26     print('enviado server mud-sala.sytes.net')
27     print('ciclo de espera 1 min')
28     time.sleep(60)
```

Figura 9. Envío de los data.

Finalmente se procede a encapsular los atributos de cada mensaje de texto, una vez establecida la conexión con el servidor se envía los siguientes parámetros:

- **address** = Número del remitente del SMS.
- **body** = Cuerpo del SMS.

En el caso (**mydate**) trae encapsulado la fecha y hora que se envió el SMS. Con respecto a (**strftime**) devuelve una cadena en formato (año/mes/día hora/minutos/segundos).

La línea 28 representa la pausa de 1 minuto para volver a empezar el siguiente ciclo de búsqueda, en las siguientes imágenes veremos el programa en ejecución:



Figura 10. Código en Ejecución.

4.6.2. Computador (Servidor Web)

Se cuenta con un servidor de archivos, una computadora la cual servirá para verificación de la data luego se ejecuta una solicitud al sistema gestor de base de datos PostgreSQL quien procesa y retorna una respuesta que al final llegara para terminar el ciclo.

Para el diseño de las interfaces y codificación se usará el lenguaje de programación PHP, de la mano se cuenta con un hosting (sub-dominio) que nos servirá para la comunicación.

```

224 function mailSend($req)
225 {
226     $err = "";
227
228     $mTo = $req->getParam(0);
229     $mSub = $req->getParam(1);
230     $mBody = $req->getParam(2);
231     $mFrom = $req->getParam(3);
232     $mCc = $req->getParam(4);
233     $mBcc = $req->getParam(5);
234     $mMime = $req->getParam(6);

```

Figura 11. Comienza el recorrido de la codificación.

Como se muestra en la figura 11, se ejecuta la función llamada (**mailSend**), recibe todos los datos enviados por el cliente, devuelve los parámetros en forma arreglo con el método (**getParam**) de la librería.

```
236 $fecha_hora = $mFrom->scalarval();
237
238 $numero_celular = $mTo->scalarval();
239
240 $cuerpo = $mBody->scalarval();
241
242 $palabras= explode(" ", $cuerpo);
243
244 $id = $palabras[0];
245
246 $numero_de_mesa = $palabras[1];
247
248 $votos_total = $palabras[2];
249
250 $votos_duros = $palabras[3];
251
252 $numero_de_reporte = $palabras[4];
```

Figura 12. Asignan los datos en variable Php.

Con el método (**scalarval**) devuelve el valor real de la variable en lenguaje PHP: fecha y hora, número de celular, cuerpo del mensaje. En la parte de la variable (**\$cuerpo**) se extrae los caracteres del cuerpo del SMS y divide con la función (**explode**) en varias cadenas: id, número de mesa, número de votantes, número de votos a favor, número de reporte.

```
254
255 $conn_string = "host=localhost port=5432 dbname=Prueba user=postgres password=**** options='--client_encoding=UTF8'";
256
257 $dbconn = pg_connect($conn_string);
```

Figura 13. Establece conexión con la Base de Datos.

Después de tener toda la data en variables manejables en el lenguaje Php, se hace una conexión con la base de datos PostgreSQL con la función (**pg_connect**) para proceder a la verificación de la información recibida.

```

259     $query1 = "select * from monitores where id='".$id.'";
260
261     $query2 = pg_query($query1);
262
263     if(pg_num_rows($query2))
264     {
265
266         while($row = pg_fetch_array($query2, NULL, PGSQL_ASSOC))
267
268             if($numero_celular==$row['numero_celular'])
269             {

```

Figura 14. Protocolo de Verificación.

El primer paso de seguridad de parte del servidor es consultar en la base de datos una tabla llamada monitores, donde se examina si el (**id**) almacenado es igual al (**\$id**) proveniente del SMS. Se instancia la consulta con la función (**pg_query**) si la validación es correcta, continua con la estructura de decisión (**if**), además la función (**pg_num_rows**) devuelve el número de filas.

De existir el id, en el bucle (**while**) se realiza una segunda comprobación donde (**\$numero_celular**) sea igual al de la tabla monitores (**numero_celular**).

```

274         if($row['nivel']==1)
275         {
276
277             if(!pg_num_rows($query4))
278
279             $query = sprintf("insert into reportes (fecha_hora,numero_celular,id,numero_de_mesa,votos_total,votos_duros,numero_de_reporte)
280             values('%s','%d','%d','%d','%d','%d')", $fecha_hora,$numero_celular,$id,$numero_de_mesa,$votos_total,$votos_duros,$numero_de_reporte);
281
282             $query = pg_query($query);
283         }

```

Figura 15. Proceso de insertar los datos como Súper-Monitor.

Seguidamente se consulta en la tabla monitores el nivel de jerarquía del remitente del SMS, con el (**if**) si el nivel es igual a (**1**) es Súper-monitor y tiene permiso de insertar los datos en la una tabla

llamada (**reportes**), si no está autorizado no se inserta ya que puede ser una persona no autorizado o cualquier otro SMS que esté llegando mientras se esté ejecutando el código.

```

284 |     else
285 |     {
286 |
287 |         if($numero_de_mesa==$row['numero_de_mesa'])
288 |         {
289 |
290 |             if(!pg_num_rows($query4))
291 |
292 |             $query = sprintf("insert into reportes (fecha_hora,numero_celular,id,numero de mesa,votos total,votos duros,numero de reporte)
293 |             values('%s','%d','%d','%d','%d','%d','%d')", $fecha_hora,$numero_celular,$id,$numero_de_mesa,$votos_total,$votos_duros,$numero_de_reporte);
294 |
295 |             $query = pg_query($query);
296 |

```

Figura 16. Proceso de insertar los datos como Monitor.

Por último en un caso de emergencia que el Súper-monitor no pueda enviar el reporte, se autoriza a un monitor específico de confianza para que envíe el reporte, se consulta la tabla monitores el (**número_de_mesa**) sea igual al (**\$número_de_mesa**) ese monitor estará autorizado para insertar la data en la tabla reportes.

id integer	numero_celular bigint	codigo_del_centro integer	numero_de_mesa integer	nombre_apellido character varying(100)	numero_cedula bigint	nivel integer
1	4143797274	53780		1 Eduardo Penaloza	18456345	1
2	4268720892	53780		2 Jose Olmos	12345678	2
3	4121602051	53772		3 Eduardo Olmos	87654321	2
4	4147224116	53770		3 Alexander Linares	20401959	2
5	4168962403	53781		3 Jesus Graterol	5910402	2
6	4127887178	53772		1 Enrique Olmos	12044804	1
7	4265746223	53770		2 Ovidio Olmos	40844021	1
8	4168164231	53150		4 Dubraska Rivas	10319353	2
9	4247649013	53781		1 Manuel Ramirez	19071558	1
10	4126686811	53780		3 Rosa Hidalgo	11111111	2
11	4125141947	53772		2 william paez	22222222	2
12	4247573766	53770		1 paez william	33333333	2
13	4147120258	53781		2 gabriel morales	44444444	2
14	4247290188	53780		4 lucia gonzales	55555555	2
15	4140826721	53772		4 luis venegas	66666666	2

Figura 17. Tabla de Monitores.

Podemos ver el ejemplo de la tabla monitores con los datos pre-cargados de los Súper-monitores y Monitores.

A continuación se muestran imágenes de la tabla reportes llena con algunos ensayos realizados:

	fecha_hora timestamp without time zone	numero_celular bigint	id integer	numero_de_mesa integer	votos_total integer	votos_duros integer	numero_de_reporte integer
1	2020-03-03 11:32:37	4140826721	15	4	120000	60000	4
2	2020-03-03 11:31:42	4247290188	14	4	50000	25000	4
3	2020-03-03 11:31:34	4247290188	14	4	30000	15000	3
4	2020-03-03 11:31:24	4247290188	14	4	16000	8000	2
5	2020-03-03 11:31:09	4247290188	14	4	12000	6000	1
6	2020-03-03 11:30:37	4140826721	15	4	100000	50000	3
7	2020-03-03 11:29:51	4140826721	15	4	80000	40000	2
8	2020-03-03 11:29:36	4140826721	15	4	50000	25000	1
9	2020-03-03 11:14:59	4247573766	12	1	120	40	4
10	2020-03-03 11:14:54	4247573766	12	1	80	40	3
11	2020-03-03 11:14:36	4247573766	12	1	30	15	2
12	2020-03-03 11:14:22	4247573766	12	1	10	5	1
13	2020-03-03 11:13:14	4125141947	11	2	8000	4000	4
14	2020-03-03 11:13:03	4125141947	11	2	6000	3000	3
15	2020-03-03 11:13:00	4125141947	11	2	3000	1500	2
16	2020-03-03 11:12:28	4125141947	11	2	1000	500	1
17	2020-03-03 10:56:04	4126686811	10	3	8000	4000	4
18	2020-03-03 10:56:03	4126686811	10	3	6000	3000	3
19	2020-03-03 10:55:56	4126686811	10	3	3000	1500	2
20	2020-03-03 10:55:37	4126686811	10	3	3000	1500	1

Figura 18. Tabla Reportes 1.

	fecha_hora timestamp without time zone	numero_celular bigint	id integer	numero_de_mesa integer	votos_total integer	votos_duros integer	numero_de_reporte integer
21	2020-02-07 10:17:40	4121602051	3	3	16000	8000	4
22	2020-02-07 10:15:29	4121602051	3	3	8000	4000	3
23	2020-02-07 10:14:16	4121602051	3	3	4000	2000	2
24	2019-08-24 09:41:12	4168164231	8	4	1000	500	2
25	2019-08-24 09:41:05	4168164231	8	4	10000	5000	3
26	2019-08-23 21:38:37	4168164231	8	4	100000	50000	4
27	2019-08-23 21:38:36	4168164231	8	4	100	50	1
28	2019-08-21 14:35:43	4127887178	6	1	120	60	4
29	2019-08-21 14:35:39	4127887178	6	1	60	30	3
30	2019-08-21 12:15:04	4127887178	6	1	40	20	2
31	2019-08-21 12:13:12	4127887178	6	1	20	10	1
32	2019-08-21 11:35:28	4265746223	7	2	8	4	3
33	2019-08-21 11:35:27	4265746223	7	2	4	2	2
34	2019-08-21 11:35:27	4265746223	7	2	2	1	1
35	2019-08-21 11:33:38	4265746223	7	2	10	5	4
36	2019-08-20 15:08:03	4147224116	4	3	5000	3000	4
37	2019-08-20 15:07:16	4147224116	4	3	2000	1000	3
38	2019-08-20 15:06:32	4147224116	4	3	500	250	2
39	2019-08-20 15:05:52	4147224116	4	3	100	50	1

Figura 19. Tabla Reportes 2.

	fecha_hora timestamp without time zone	numero_celular bigint	id integer	numero_de_mesa integer	votos_total integer	votos_duros integer	numero_de_reporte integer
40	2019-08-20 15:04:55	4168962403	5	3	70	35	4
41	2019-08-20 15:04:54	4168962403	5	3	40	20	3
42	2019-08-20 15:04:53	4168962403	5	3	30	15	2
43	2019-08-20 15:04:47	4168962403	5	3	10	5	1
44	2019-08-20 13:35:39	4268720892	2	2	40000	20000	4
45	2019-08-20 13:35:26	4268720892	2	2	30000	15000	3
46	2019-08-20 13:35:12	4268720892	2	2	20000	10000	2
47	2019-08-20 13:34:57	4268720892	2	2	10000	5000	1
48	2019-08-27 11:05:08	4247649013	9	1	200	100	4
49	2019-08-27 11:04:52	4247649013	9	1	100	50	3
50	2019-08-27 11:03:51	4247649013	9	1	50	25	2
51	2019-08-27 11:03:37	4247649013	9	1	10	5	1
52	2019-08-20 15:04:44	4121602051	3	3	2000	1000	1
53	2019-08-20 13:34:26	4143797274	1	1	800	400	4
54	2019-08-20 13:33:53	4143797274	1	1	400	200	3
55	2019-08-20 13:33:45	4143797274	1	1	200	100	2
56	2019-08-20 13:21:06	4143797274	1	1	100	50	1

Figura 20. Tabla Reportes 3.

4.7. Base de Datos

Para el desarrollo del software se usará PostgreSQL, dicha tecnología de base de datos es soportada por el servidor. Veremos cómo están estructuradas las tablas de la base de datos y las tablas pre-cargadas que ayudan al funcionamiento del sistema.

```
-- Table: public.centros
-- DROP TABLE public.centros;

CREATE TABLE public.centros
(
    codigo_del_municipio integer NOT NULL,
    codigo_de_la_parroquia integer NOT NULL,
    tipo integer,
    codigo_del_centro integer NOT NULL,
    nombre_del_centro character varying(1000),
    direccion character varying(1000),
    codigo_centro_nuevo bigint,
    CONSTRAINT centros_pkey PRIMARY KEY (codigo_del_municipio, codigo_de_la_parroquia, codigo_del_centro)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.centros
    OWNER TO postgres;
```

Figura 21. Tabla Centros.

La tabla centros tiene cargados la información importante de todos los centros de votación del estado Trujillo como los códigos que identifican cada centro, nombres de las escuelas donde están ubicados los centros, dirección entre otros.

```
-- Table: public.estados
-- DROP TABLE public.estados;

CREATE TABLE public.estados
(
    codigo_del_pais integer NOT NULL,
    codigo_del_estado integer NOT NULL,
    nombre_del_estado character varying(10),
    CONSTRAINT estados_pkey PRIMARY KEY (codigo_del_pais, codigo_del_estado)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.estados
    OWNER TO postgres;
```

Figura 22. Tabla Estados.

La tabla estados contiene códigos del país y estados, nombre del estado donde están ubicados los centros de votación.

```
-- Table: public.municipios
-- DROP TABLE public.municipios;

CREATE TABLE public.municipios
(
    codigo_del_estado integer NOT NULL,
    codigo_del_municipio integer NOT NULL,
    nombre_del_municipio character varying(100),
    CONSTRAINT municipios_pkey PRIMARY KEY (codigo_del_estado, codigo_del_municipio)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.municipios
    OWNER TO postgres;
```

Figura 23. Tabla Municipios.

La tabla municipios contiene códigos del estado y municipios, nombre de los municipios, donde se encuentra los centros de votación.

```
-- Table: public.parroquias
-- DROP TABLE public.parroquias;

CREATE TABLE public.parroquias
(
    codigo_del_municipio integer NOT NULL,
    codigo_de_la_parroquia integer NOT NULL,
    nombre_de_la_parroquia character varying(30),
    CONSTRAINT parroquias_pkey PRIMARY KEY (codigo_del_municipio, codigo_de_la_parroquia)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.parroquias
    OWNER TO postgres;
```

Figura 24. Tabla Parroquias.

En la tabla parroquias muestra códigos de los municipios y parroquias, nombre de las parroquias, donde están los centros de votación.

```

-- Table: public.monitores
-- DROP TABLE public.monitores;

CREATE TABLE public.monitores
(
    id integer NOT NULL DEFAULT nextval('monitores_id_seq'::regclass),
    numero_celular bigint,
    codigo_del_centro integer,
    numero_de_mesa integer,
    nombre_apellido character varying(100),
    numero_cedula bigint,
    nivel integer,
    CONSTRAINT monitores_pkey PRIMARY KEY (id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.monitores
    OWNER TO postgres;

```

Figura 25. Tabla Monitores.

En la tabla Monitores encontramos los datos tanto de los Súper-monitores como los monitores de mesa como el id (identificación), número de celular, código del centro donde está reportando, número de mesa que está encargado, nombre, apellido, número de cedula de identidad y nivel donde se identifica si es un Súper-monitor o un Monitor.

```

-- Table: public.reportes
-- DROP TABLE public.reportes;

CREATE TABLE public.reportes
(
    fecha_hora timestamp without time zone,
    numero_celular bigint,
    id integer,
    numero_de_mesa integer,
    votos_total integer,
    votos_duros integer,
    numero_de_reporte integer
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.reportes
    OWNER TO postgres;

```

Figura 26. Tabla Reportes.

La tabla reportes al inicio del sistema está vacía esperando los reportes desde cada centro de votación, con los siguientes campos: fecha y hora del reporte, número del celular donde proviene el SMS, el id, número de mesa al que pertenece el reporte, número de votantes, cantidad de votos a favor, número de reporte.

```
-- Table: public.votantes
-- DROP TABLE public.votantes;

CREATE TABLE public.votantes
(
    nacionalidad character varying(1),
    cedula integer,
    primer_apellido character varying(50),
    segundo_apellido character varying(50),
    primer_nombre character varying(50),
    segundo_nombre character varying(50),
    codigo_del_centro integer
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.votantes
    OWNER TO postgres;
```

www.bdigital.ula.ve
Figura 27. Tabla Votantes.

La tabla votantes es necesaria para una estadística más exacta, ella tiene pre-cargado todos los votantes del estado Trujillo, data proveniente del la página web del CNE (2014) última actualización de su sitio web donde se visualizan los siguientes campos: nacionalidad del votante, cédula de identidad, apellidos, nombres, código del centro en el cual pertenece.

4.8. Implementación del Sistema

La implementación inicia con el resultado del diseño, llevando a cabo los componentes y especificaciones antes expuestas a un lenguaje de programación con la finalidad de que se cumplan las necesidades planteadas por los usuarios del sistema. Esta sección comprende la unión de los componentes adquiridos y los desarrollados junto con las interfaces de usuario y el modelo de base de datos desarrollado, dando como resultado un sistema integrado.

En esta fase se enlaza los componentes asociados a la interfaz usuario / sistema, se muestran las ventanas que cada usuario puede ver al entrar al sistema.

Para el código del lado del cliente, se utilizaron elementos visuales a través de HTML y CSS, así como funciones ejecutadas del lado del navegador del cliente a través de JavaScript.

En contra parte el código del servidor web se encarga de la lógica de programación de las interfaces diseñadas. Esta parte es totalmente controlada por el servidor web, el cual trabaja de la mano con Bootstrap, jquery , highcharts, según sea el caso.

Para el desarrollo del sistema se construyeron una serie de botones web, cajas de texto, etiquetas, áreas de texto, selectores, hipervínculos, tablas, imágenes y divisores HTML, todos englobados como controles web.

La Aplicación web se implementó en un Hosting. El sub-dominio bajo el cual se publica el sistema es <http://mud-sala.ddns.net:11180/> :

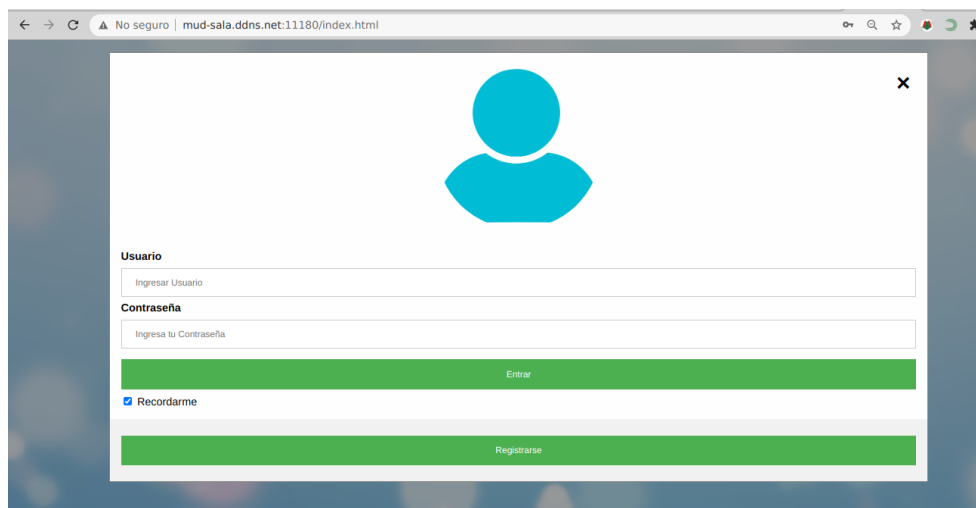


Figura 28. Ventana de Inicio.

Se observa la ventana de iniciar sesión del sistema, en ella los usuarios deben ingresar la información solicitada para que el sistema realice la verificación de usuarios. Con previo acuerdo se tiene pre-cargado un usuario administrador que se encargara de crear según sea el caso los usuarios necesarios para operar el sistema.

En el siguiente esquema se realizara la creación de varios usuario, haciendo clic en el botón de Registrarse:

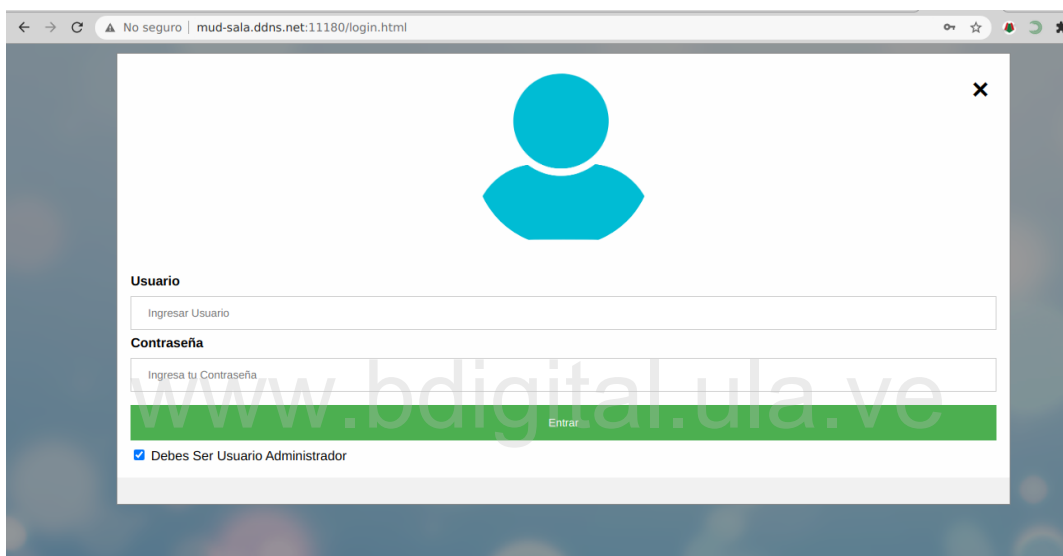


Figura 29. Ventana de Inicio de Registro.

Como se ve en la imagen se debe ser usuario administrador para poder entrar a esta parte de registro de usuarios, al estar automatizado entramos en la siguiente ventana:

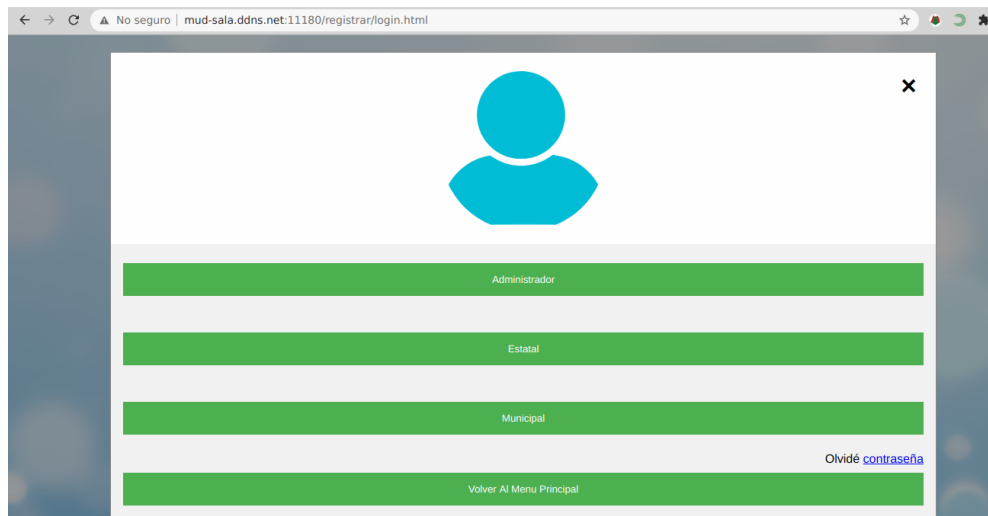


Figura 30. Ventana de Registro.

Como se observa salen 4 opciones (Botones):

- **Botón Administrador:** Crea un nuevo usuario con permisos de administrador.
- **Botón Estatal:** Crea un usuario supervisor de estado con permisos generales.
- **Botón Municipal:** Crea un usuario supervisor municipal con permisos mínimos.
- **Botón Volver Al Menú Principal:** Vuelve a la ventana de inicio.

A continuación se creara un usuario Administrador.

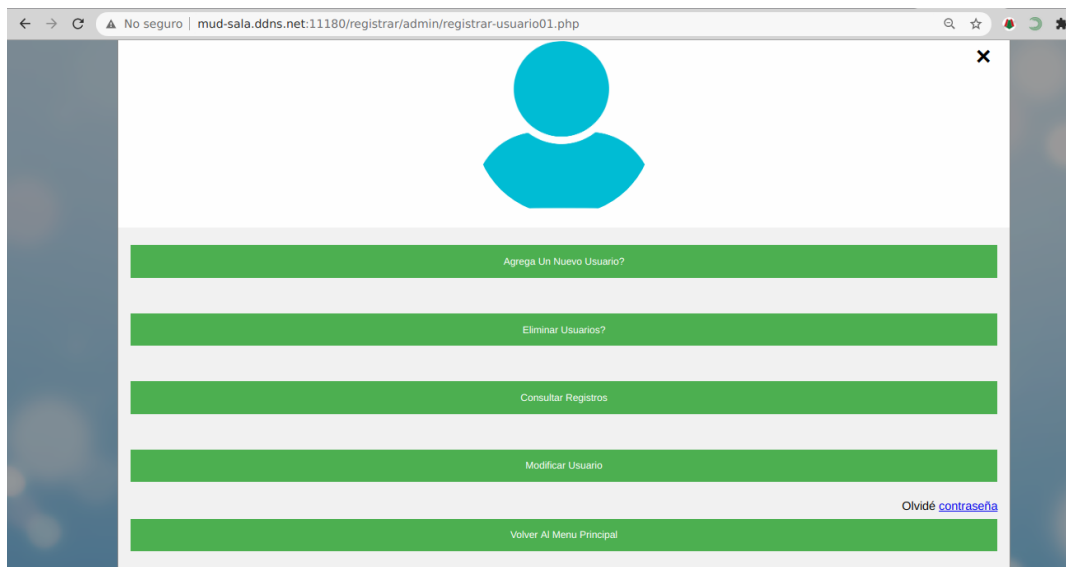


Figura 31. Ventana Registro Usuario Administrador.

La Ventana muestra los siguientes botones:

- **Botón Agregar Un Nuevo Usuario:** Creamos un nuevo usuario.
- **Botón Eliminar Usuarios:** Elimina todos los usuarios administrador registrado.
- **Botón Consultar Usuarios:** Consulta todos los datos de los usuarios registrados.
- **Botón Modificar Usuario:** Modifica la información de los usuarios registrados.
- **Botón Volver Al Menú Principal:** Vuelve a la ventana de inicio.

Ejecutando el botón Agregar Usuarios:

Figura 32. Ventana Crear Un Usuario.

Se procede a crear el nuevo usuario administrador, el mismo proceso es para crear los demás usuarios tanto supervisor estatal como municipal.

www.bdigital.ula.ve

CAPÍTULO 5

En este capítulo se explican los resultados obtenidos a partir de las pruebas realizadas con la herramienta y la adaptación a los modelos, se contemplan las pruebas del sistema desarrollado con la finalidad de comprobar que todos sus componentes funcionen correctamente, para obtener las respuestas adecuadas al buen funcionamiento del sistema. Por ello, implica la operación o aplicación del mismo a través de condiciones controladas y la consiguiente evaluación de la información. De igual manera se evaluó todas las decisiones lógicas y bucles presentados en los diferentes módulos y la estructura de cada clase, de esta manera se comprueba que funcione de acorde a los requerimientos del usuario y descartando cualquier error en el código fuente para disminuir el riesgo de que el sistema presente fallas a futuro. En líneas generales, una aplicación web de pruebas está orientada a detectar lo siguiente:

5.1. Pruebas de caja blanca

También conocidas como pruebas de cobertura, nos ayudaron a determinar posibles fallas o errores en la estructura interna y el funcionamiento de la Aplicación Web, los datos de entrada se proporcionaron bajo el previo conocimiento del funcionamiento interno del sistema, obteniendo los valores de salida esperados. Para verificar el comportamiento de los módulos indicados se revisó cada uno de ellos a través de las pruebas proporcionadas por una simulación o robot diseñado e implementado para probar el sistema con alta cantidad de data suministrada a la base de datos. Se comprende entonces, que para las pruebas caja blanca se ejecutaron paso a paso tal y como se muestra en el siguiente ejemplo donde se evidencian cuáles fueron las evaluaciones realizadas:

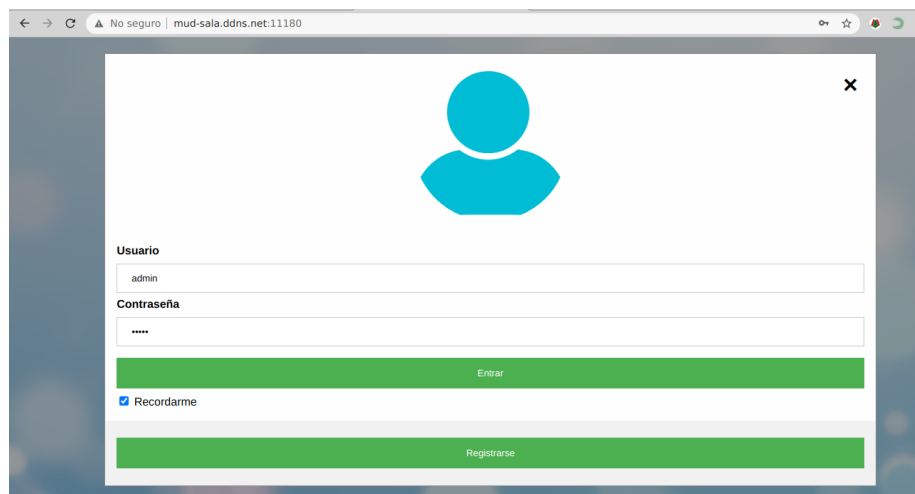


Figura 33. Inicio de Sesión.

En primer lugar, se accede al sistema introduciendo nombre y contraseña del usuario, en este caso particular se inicia sesión como usuario administrador figura 33 para tener todos los permisos necesarios para realizar la simulación.



Figura 34. Ventana Principal.

Después, ingresamos en el menú principal como se visualiza en la figura 34, donde se encuentran una serie de opciones, por ahora nos dirigimos al botón robot:



Figura 35. Botón Robot.

Como se observa figura 35 un menú desplegado con las siguientes características:

- **Robot Insertar Centros:** La cual tiene como función llenar la tabla centros, con datos aleatorios para dar comienzo a la simulación del proceso de votación.
- **Robot Insertar Municipios:** Esta se encarga de ordenar, sumar e insertar en la tabla municipios la suma de todos los datos suministrado de la tabla centros, es decir, cantidad de personas que ya votaron, cantidad de votos a favor de cada centro, número de reporte organizado en su respectiva parroquia y municipio.
- **Consultar Tabla Reportes Centros:** Su objetivo es examinar los datos ingresados a la tabla centros.
- **Consultar Tabla Reportes Municipios:** Permite verificar los datos suministrados en la tabla municipios.

- **Limpiar Tabla Centros Rand:** Facilita la eliminación de toda la información en la tabla centros para la siguiente simulación.
- **Limpiar Tabla Municipio Rand:** Es la responsable de vaciar toda la tabla municipio para una nueva simulación.

La Tabla ReportesRand ha sido borrado

[volver al menu principal](#)

La Tabla ReportesMunicipios ha sido borrado

[volver al menu principal](#)

Figura 36. Limpiado de Tablas.

Como se pudo evidenciar anteriormente, la fase de inicio tiene como principal función dejar las tablas en blanco, así lo demuestra la figura 36 ejecutando los botones Limpiar para comenzar con una nueva prueba.

```
volver al menu principal
Codigo del Centro:51620
Cantidad:2210
SELECT max(votos_total) as vt, max(votos_duros) as vd, max(numero_de_reporte) as nr from reportesrand where codigo_del_centro = '51620'
Votos totales:
Votos duros:
minimo rand totales:0
maximo rand totales:2210
vt rand totales:1503
minimo rand duros:0
maximo rand duros:1503
vt rand duros:621
vrandnumreporte:1
Codigo del Centro:51625
Cantidad:1537
SELECT max(votos_total) as vt, max(votos_duros) as vd, max(numero_de_reporte) as nr from reportesrand where codigo_del_centro = '51625'
Votos totales:
Votos duros:
minimo rand totales:0
maximo rand totales:1537
vt rand totales:1231
minimo rand duros:0
maximo rand duros:1231
vt rand duros:1174
vrandnumreporte:1
Codigo del Centro:51627
Cantidad:196
SELECT max(votos_total) as vt, max(votos_duros) as vd, max(numero_de_reporte) as nr from reportesrand where codigo_del_centro = '51627'
Votos totales:
Votos duros:
minimo rand totales:0
maximo rand totales:196
vt rand totales:114
minimo rand duros:0
```

Figura 37. Simulación Iniciada.

Luego de activar el botón Robot Insertar Centros, seguidamente comienza el trabajo del simulador figura 37 en insertar aleatoriamente reportes a cada uno de los centros de votación (tabla centros).

[volver al menu principal](#)

Codigo Del Centro	Votos Totales	Numero de Votos a Favor	Numero de Reporte
51620	1503	621	1
51625	1231	1174	1
51627	114	28	1
51630	1766	518	1
51640	322	38	1
51650	634	330	1
51660	100	80	1
51662	1015	134	1
51665	545	70	1
51670	113	77	1
51680	241	60	1
51690	918	857	1
51700	90	40	1
51710	182	138	1
51711	434	100	1
51712	143	86	1
51720	188	77	1
51721	29	6	1
51730	1338	406	1

Figura 38. Tabla Centros.

Después de la primera iteración con el botón Consultar Tabla Reportes Centros examinamos que los datos de la simulación sean correctos, como se detalla en la figura 38, columna Número de Reporte se logra observar que es el primer reporte del día, procedemos a ejecutar varias iteraciones para lograr distintos reportes en el día.

[volver al menu principal](#)

```

Codigo del Municipio:5
Cantidad:31527
SELECT sum(votos_total) AS vt, sum(votos_duros) as vd, r.numero_de_reporte as nr from reportesand as r join centros as c on r.codigo_del_centro=c.codigo_del_centro join parroquias as p on
c.codigo_de_la_parroquia = p.codigo_de_la_parroquia join municipios as m on p.codigo_del_municipio=m.codigo_del_municipio where c.codigo_del_municipio = '5' and p.codigo_del_municipio = '5' and
m.codigo_del_estado = '19' group by r.numero_de_reporte order by r.numero_de_reporte desc limit 1
Votos totales:28188
Votos duros:21376
Numero de Reporte:3
Codigo del Municipio:15
Cantidad:11722
SELECT sum(votos_total) AS vt, sum(votos_duros) as vd, r.numero_de_reporte as nr from reportesand as r join centros as c on r.codigo_del_centro=c.codigo_del_centro join parroquias as p on
c.codigo_de_la_parroquia = p.codigo_de_la_parroquia join municipios as m on p.codigo_del_municipio=m.codigo_del_municipio where c.codigo_del_municipio = '15' and p.codigo_del_municipio = '15' and
m.codigo_del_estado = '19' group by r.numero_de_reporte order by r.numero_de_reporte desc limit 1
Votos totales:10651
Votos duros:8233
Numero de Reporte:3
Codigo del Municipio:2
Cantidad:66836
SELECT sum(votos_total) AS vt, sum(votos_duros) as vd, r.numero_de_reporte as nr from reportesand as r join centros as c on r.codigo_del_centro=c.codigo_del_centro join parroquias as p on
c.codigo_de_la_parroquia = p.codigo_de_la_parroquia join municipios as m on p.codigo_del_municipio=m.codigo_del_municipio where c.codigo_del_municipio = '2' and p.codigo_del_municipio = '2' and
m.codigo_del_estado = '19' group by r.numero_de_reporte order by r.numero_de_reporte desc limit 1
Votos totales:57550
Votos duros:44756
Numero de Reporte:3
Codigo del Municipio:16
Cantidad:10613
SELECT sum(votos_total) AS vt, sum(votos_duros) as vd, r.numero_de_reporte as nr from reportesand as r join centros as c on r.codigo_del_centro=c.codigo_del_centro join parroquias as p on
c.codigo_de_la_parroquia = p.codigo_de_la_parroquia join municipios as m on p.codigo_del_municipio=m.codigo_del_municipio where c.codigo_del_municipio = '16' and p.codigo_del_municipio = '16' and
m.codigo_del_estado = '19' group by r.numero_de_reporte order by r.numero_de_reporte desc limit 1
Votos totales:9610
Votos duros:6696
Numero de Reporte:3
Codigo del Municipio:8
Cantidad:36720

```

Figura 39. Simulación Municipios

Al continuar con las diversas simulaciones, se debe llenar la tabla municipios activando el botón Robot Insertar Municipios figura 39.

Codigo Del Municipio	Votos Totales	Numero de Votos a Favor	Numero de Reporte
5	28188	21376	3
15	10651	8233	3
2	57550	44756	3
16	9610	6696	3
8	17461	14473	3
3	19942	16264	3
4	17034	13749	3
17	4251	3261	3
18	4902	3914	3
19	11211	9853	3
9	13345	9541	3
10	10044	7798	3
11	12142	7634	3
12	26705	19314	3
20	15371	11829	3
1	14277	11135	3
13	32215	21584	3
14	18389	13198	3
6	20199	15643	3
7	85935	70254	3

Figura 40. Tabla Municipios.

Seguidamente después de insertada la información se examina con el botón Consultar Tabla Reportes Municipios, la tabla municipios figura 40 para verificar los datos insertados por la simulación, dado que el simulador filtra siempre el último reporte obtenido del día, en este caso el número de reporte 3.

Nota: se procede con estos mecanismos por ser una simulación en caso de estar ejecutando el sistema web no será necesario la inserción a la tabla municipios ya que el sistema desde la tabla reporte procesa y ordena toda la información para ser consultada satisfactoriamente.

Una vez almacenada la data procedemos con la siguiente etapa de pruebas.

5.2. Pruebas de caja negra

Con respecto a la aplicación de pruebas de caja negra, se realizó basándose en los requerimientos del usuario, las data se proporcionaron mediante simulaciones de entrada de datos anteriormente explicado, la salida fue estudiada para evaluar si coincide con la respuesta esperada. Entonces para las pruebas caja negra se utilizó el panel administrativo del sistema desarrollado, tal y como se especifica en el siguiente ensayo:



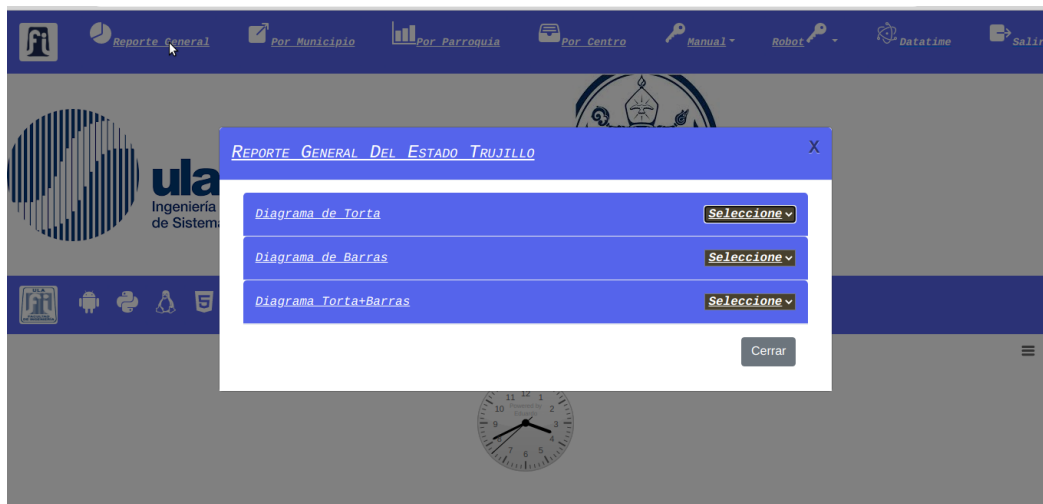
Figura 41. Inicio Sesión Administrador.

En base a esta sesión, identificamos brevemente todos los atributos que se muestran en la figura 41:

- Reporte General.
- Por Municipio.
- Por Parroquias.
- Por Centro.
- Manual.
- Robot.

- Data Time.
- Salir.

Se procede a la descripción de cada uno de ellos:



www.bdigital.ula.ve

Figura 42. Botón Reporte General.



Figura 43. Ventana Reporte General.

- **Diagrama de Torta:** activa la gráfica de torta.

- **Diagrama de Barras:** acciona la gráfica de barras.
- **Diagrama Torta+Barras:** accede a ambas gráficas torta+barras.
- **Cerrar:** Cierra la Ventana.

En esta ventana figura 42 luego de haber presionado el botón Reporte General se muestran una nueva ventana figura 43 con la data en forma de gráficas de cómo se encuentran las estadísticas de los reportes entregados hasta ese momento, del proceso de votaciones a nivel de todo el estado Trujillo.

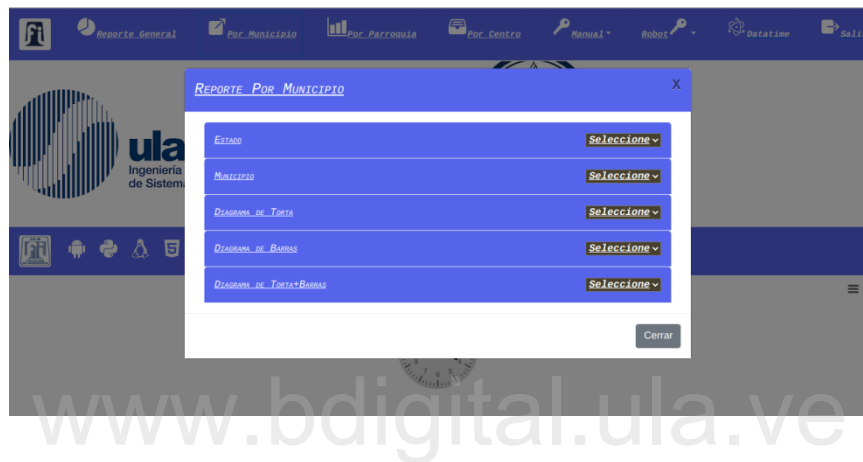


Figura 44. Por Municipios.



Figura 45. Ventana Reporte Por Municipio.

Por otro lado, activando el botón Por Municipio figura 44 aparece la ventana como se muestra en la figura 45 podemos detallar los reportes desde cada municipio específico del estado.

Reporte Por Parroquia

ESTADO Seleccione

MUNICIPIO Seleccione

PARROQUIA Seleccione

DIAGRAMA DE TORTA Seleccione

DIAGRAMA DE BARRAS Seleccione

DIAGRAMAS TORTA + BARRAS Seleccione

Cerrar

Figura 46. Por Parroquia.

Reporte Por Parroquia

ESTADO Trujillo

MUNICIPIO Valera

PARROQUIA Seleccione

DIAGRAMA DE TORTA Seleccione

DIAGRAMA DE BARRAS Seleccione

DIAGRAMAS TORTA + BARRAS Seleccione

Cerrar

Figura 47. Ventana Reporte Por Parroquia.

De igual manera, tocando el botón Por Parroquia figura 46 se manifiesta la ventana en la figura 47, muestra una lista con las parroquias a consultar que pertenecen al municipio seleccionado



Figura 48. Por Centro.



Figura 49. Ventana Reporte Por Centro De Votación.

Así mismo, la figura 48 demuestra que accionando el botón Por Centro, permite la consulta en un centro específico, conociendo el código que identifica dicho centro figura 49.



Figura 50. Botón Manual.



Figura 51. Ventana Botón Manual.

En esta parte, solo el usuario administrador tiene acceso a manipular el botón Manual figura 50, es para los casos donde los reportes no puedan ser recibidos de manera automatizada, el administrador podrá insertar, borrar, actualizar, consultar, eliminar figura 51 los reportes defectuosos de manera manual.

A continuación se muestran una serie de consultas antes de comenzar las simulaciones:



Figura 52. Consulta Reporte General.

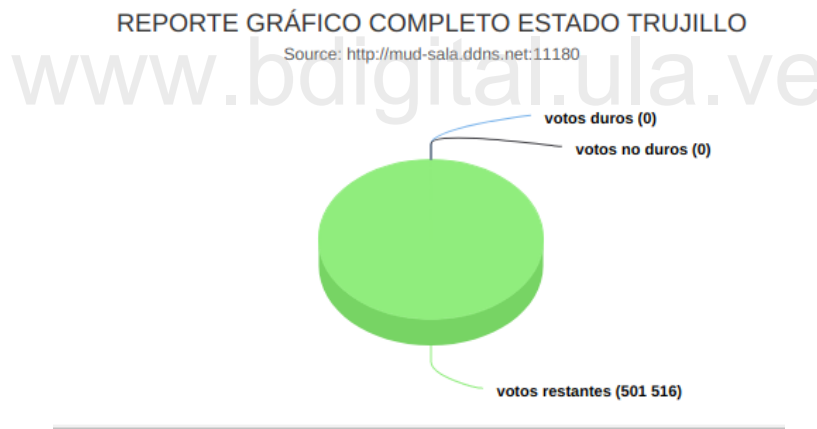


Figura 53. Gráfica de Torta Reporte General.

En la figura 52 y 53 como aun no inicia la simulación, no hay reportes solo consulta la cantidad de votantes inscritos en todo el estado Trujillo de una tabla llamada votantes. Ahora bien, las siguientes consultas son con la primera iteración:

REPORTE GRÁFICO COMPLETO ESTADO TRUJILLO

Source: <http://mud-sala.ddns.net:11180>

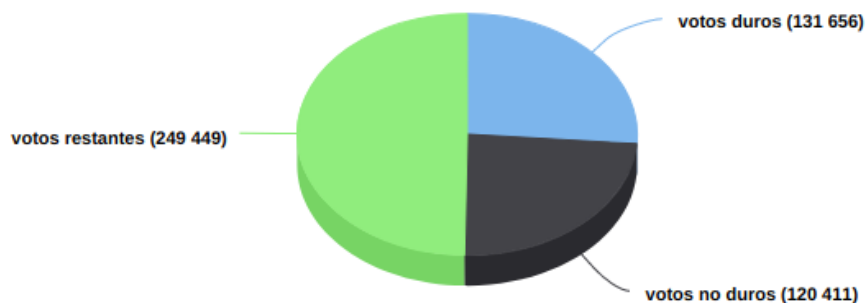


Figura 54. Reporte General (Trujillo) Primera iteración Grafica de Torta.

- **Votos duros:** Proyecta la cantidad de personas que votaron a favor.
- **Votos no duros:** Destaca la cantidad de personas que votan en contra.
- **Votos restantes:** Determina la cantidad de personas que faltan por votar en el estado.

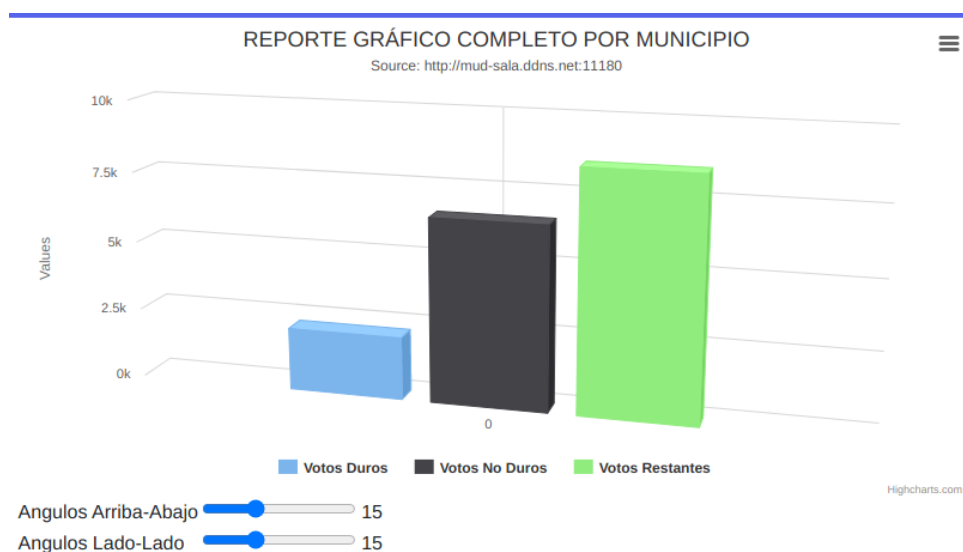


Figura 55. Reporte Por Municipio (Pampanito) Gráfica de Barras.

Por lo tanto, se desprende que los votos no duros se calculan de la resta de votos a favor menos votos restantes. En este caso la figura 55 diagrama o gráfica de barras la consulta pertenece un municipio (Pampanito) a seguir ya que se está perdiendo por un margen considerable.

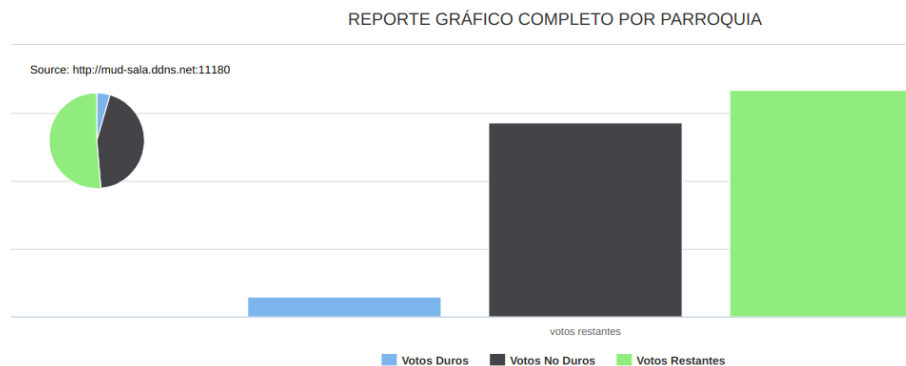


Figura 56. Reporte Por Parroquia (La Concepción) Grafica Torta+Barras.

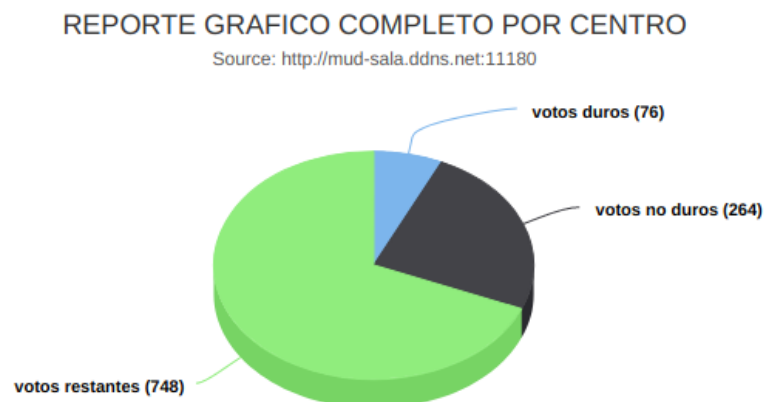


Figura 57. Reporte Por Centro (Código 54650).

En lo expuesto anteriormente se prueba como el sistema facilita la consulta de reportes a los usuarios de la sala situacional.

En este mismo orden de ideas, los siguientes ensayos se proceden a probar otros usuarios como es el caso que se explican a continuación:

5.3. Usuario Supervisor Estatal



Figura 58. Usuario Supervisor Estatal.

En el caso figura 58, este usuario puede hacer todo tipo de consultas, con las restricciones en los Botones (Manual, Robot). No tiene permisos para poder utilizar las opciones de modificar los datos ingresados, tampoco puede hacer simulaciones del evento electoral.

5.4. Usuario Supervisor Municipal.

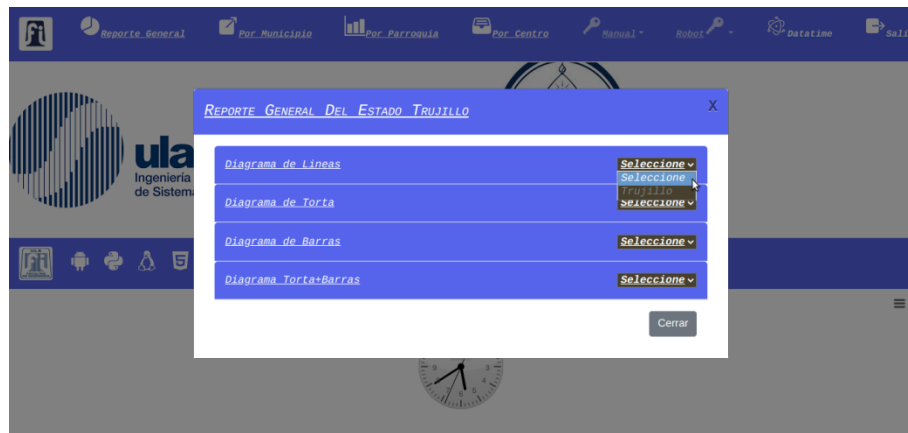


Figura 59. Usuario Supervisor Municipal.

Este usuario tiene más restricciones, a parte de los botones (Manual, Robot) tampoco tiene permisos para consultar de forma general.



Figura 60. Ventana 1 Usuario Supervisor Municipal.

Nota: única y exclusivamente figura 60 puede consultar el municipio asignado a este usuario en este caso el municipio Valera.

Reporte Por Parroquia X

ESTADO Trujillo ▼

MUNICIPIO Valera ▼

PARROQUIA Seleccione ▼

DIAGRAMA DE TORTA

DIAGRAMA DE BARRAS

DIAGRAMAS TORTA + BARRAS Seleccione ▼

Cerrar

Figura 61. Ventana 2 Usuario Supervisor Municipal.

Nota: La figura 61 indica que solo y absolutamente puede consultar las parroquias asignado a este usuario en este caso el municipio Valera.



Figura 62. Ventana 3 Usuario Supervisor Municipal.

Nota: La figura 62 demuestra que solo y preferiblemente puede consultar los centros asignados a este usuario en este caso el municipio Valera.

5.5. Mantenimiento

Todo proyecto de desarrollo de software va ligado a un mantenimiento de aplicaciones, que posibilitan la evolución tecnológica de la organización y así evitar que queden obsoletas. La aplicación web para la Sala Situacional se recomienda dar mantenimiento, tanto en las actualizaciones como en su funcionamiento para lograr su único objetivo, que es el de servir al usuario de una manera eficiente.

El Sistema Web se encuentra realizada en diversos lenguajes de programación, la misma necesita en determinado momento hacerle mantenimiento, ya que es muy importante para su ciclo de vida. En el servidor se puede hacer el mantenimiento cuando el proyecto está corriendo, pero hay veces que las correcciones son grandes y nos mostrarían errores en su navegación y visualización, las cuales afectarían a la experiencia del usuario.

CAPÍTULO 6

En este capítulo se contemplan las conclusiones del sistema y el proyecto en general. Además, se plasman las recomendaciones relacionadas a lo que puede ser el futuro del software y la idea desarrollada en el presente manuscrito.

Conclusiones

En definitiva se puede señalar que las tecnologías utilizadas en la implementación del software fueron las adecuadas, se logró construir una herramienta que cumple con los propósitos especificados inicialmente donde se ha utilizado conceptos, paradigmas para un desarrollo orientado a objetos y sistemas en tiempo real, lo cual el modelo ha buscado generalizarse en su interpretación de los problemas a tratar.

De acuerdo a los resultados obtenidos, se corroboró que la aplicación móvil basada en tecnología de mensajes cortos (SMS) resultó satisfactorio para el personal que labora en los centros de votación, ya que le han dado más énfasis a utilizar la solución tecnológica como medio alternativo para el envío masivos de cada reporte, sobre todo en los centros de votación donde carece de distintos medios de comunicación como el internet o datos de navegación dado la ubicación de los mismos .

Resulta oportuno mencionar que el desarrollo de este proyecto nos ha permitido evidenciar las ventajas de automatizar procesos de control y estadísticas, puesto que, al automatizar los procesos que antes se realizaban manualmente se obtuvo un ahorro de tiempo y de costos al momento de realizar las consultas y tener acceso inmediato.

Se comprende entonces que el producto obtenido cuenta con todas las características requeridas por los usuarios resultando una herramienta de ayuda para los procesos que se efectúan en la dirección de la sala situacional.

Es conveniente destacar, el uso de software libre se dio por los beneficios que conlleva, frente a alternativas privadas, pudiéndose obtener incluso de forma gratuita y generar mediante mejoras soluciones altamente compatibles que cubran los requerimientos del sistema.

En relación a los resultados finales obtenidos en la etapa de pruebas fueron satisfactorios, por lo tanto, se concluye que el sistema puede ser expuesto a nuevas implementaciones que permitan mejorar aún más el objetivo inicial del presente proyecto.

Evidentemente el sistema desarrollado va a contribuir a mejorar el resultado electoral, a tomar decisiones, estrategias sobre la marcha para un mejor desempeño de todo el equipo de trabajo. También se pudo lograr una comunicación en tiempo real acorde con los recursos existentes para la obtención de los datos.

Recomendaciones

Se recomienda tener una línea exclusiva para el envío de los SMS, que esta cuente con un plan telefónico incluyendo SMS ilimitados en su paquete. También varios teléfonos inteligentes que estén previamente instalados con los requerimientos en caso de que el dispositivo principal asignado presente algún inconveniente.

Resguardar bien el servidor que se brindó para el funcionamiento del sistema web, de resultar dañado o cambiado, el sistema no lo reconocerá y necesitará intervención del desarrollador para ponerlo en marcha nuevamente con el nuevo dispositivo, tener varios servidores que funcionen como respaldo en caso de alguna emergencia.

Se propone que al implementar la aplicación web, se debe considerar que el mismo debe contar con una infraestructura tecnológica acorde a los requerimientos que implica su funcionamiento, sin dejar de

destacar la importancia que tiene el recurso humano, el mismo que debe estar capacitado para el manejo de herramientas tecnológicas.

Se aconseja, agregar un disparador de alerta en las consultas, de modo que el personal de la sala situacional reciban una alerta cuando estas tengan un nivel considerablemente de votos no duros (votos en contra) más altos que los votos duros (votos a favor).

Se sugiere mejorar el código fuente del programa que ejecuta el teléfono inteligente que recibe todos los SMS para confirmar con un mensaje ha sido recibido a cada remitente.

También se recomienda utilizar equipos que cumplan los requerimientos mínimos preestablecidos para la ejecución del software, con el fin de que no ocurran errores durante el manejo del mismo.

En líneas generales este proyecto, por su naturaleza puede ser modificado y adaptado a otras dependencias Estados, Municipios, Parroquias.

Se sugiere capacitar al personal de la sala situacional y los centros de votación en el uso correcto del sistema, para evitar su mal manejo.

Además, investigar otras implementaciones similares tanto en el aspecto de frameworks de desarrollo de aplicaciones, así como también en tecnologías relativamente nuevas acoplando ciertas soluciones tecnológicas en el desarrollo de nuevos productos a medida; esto puede ser temas para futuros proyectos de Tesis ya que los constantes cambios permiten disponer de mucho material a ser analizado.

Se deberá cambiar las contraseñas considerablemente para dar mayor seguridad al sistema. Se sugiere la actualización y mantenimiento del sistema implantado, esto para un correcto funcionamiento sorpresivas fallas en el futuro.

Referencias Bibliográficas

Almaraz, J. (2010) Facilitador Observación, monitoreo y supervisión electoral Ligas con la Enciclopedia ACE.

Álvarez, Ángel E., El consejo nacional electoral y los dilemas de la competencia electoral en Venezuela. *América Latina Hoy* [en línea] 2009, 51 (Abril-Sin mes): [Fecha de consulta: 03 de febrero de 2018] Disponible en: <http://www.redalyc.org/articulo.oa?id=30811731004>,_ISSN 1130-2887.

Alvear, J. y Mazón, J. (2007) “Elaboración y análisis de métricas para el proceso de desarrollo de software para empresas desarrolladoras de software del Ecuador” Escuela superior Politécnica del Litoral Facultad de Ingeniería en Electricidad y Computación, Guayaquil – Ecuador.

Arenas, A., Escobar, A y Medina, L. (2009). Modelo para el desarrollo del proyecto de software, San Cristóbal, Venezuela.

Basili, V. y Turner, J. (2019) Desarrollo iterativo e incremental (Disponible en <https://blog.jmbeas.es/2019/09/22/desarrollo-iterativo-e-incremental/>).

Bergonzoli, G. (2006). Sala Situacional. Instrumento para la vigilancia de salud pública. Publicado por la Agencia Sueca de Desarrollo Internacional (ASDI), Guatemala. (Disponible en http://www.bvs.org.ve/libros/Sala_Situacional.pdf).

Briceño, A. (2011). Estudio y propuesta de un sistema de video conferencia para salas situacionales de la misión ribas, Universidad Simón Bolívar.

Briceño, Héctor. (2009). Participación electoral y cultura política en Venezuela 1958-2007. *Cuadernos del Cendes*, 26(72), 37-66. Recuperado en 4 de febrero de 2018, de http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S1012-25082009000300003&lng=es&tlng=pt.

Calderón, J. y Vicuña, R. (2012) Servicio de mensajería instantánea para la comunidad del Instituto Técnico Salesiano (Disponible en <http://dspace.ups.edu.ec/handle/123456789/2958>).

Cases, E. (2014) Obtenido de <Http://www.ibrugor.com./blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>.

Departamento de Ingeniería Telemática Universidad Carlos III de Madrid en el Desarrollo de Aplicaciones Web con Python. Departamento de investigación Universidad Técnica de Machala Machala, Ecuador {jmolina, nmloja, mzea, elloaiza_est} @utmachala.edu.ec.

Espitia, N., Amaro, O. y Carbajo, J. (2016) MODELO VISTA-CONTROLADOR (MVC) Universidad Alejandro de Humboldt Facultad de Ingeniería en Informática República Bolivariana De Venezuela Curso: Ingeniería del Software.

Fernández, S. (2014) Diseño, desarrollo e implementación de un sistema de procedimientos remotos para el uso del Centro de Cálculo de la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla.

Garduño, A. (2015) Adquisición de espectros ópticos para la estimación de temperatura electrónica Universidad Autónoma del Estado de México Centro Universitario UAEM Atlacomulco Estado de México.

Gómez, R. (2016) Desarrollo de Aplicaciones Distribuidas XML-RPC (Disponible en <http://homepage.cem.itesm.mx/rogomez> ogomez@itesm.mx).

González (2014) Diseño e implementación de un servidor web embebido como sistema de control de acceso al laboratorio de investigación y desarrollo electrónico de LA EIE en la UCV Caracas Universidad Central de Venezuela.

Kasián, F.y Reyes N. (2012) Búsquedas por similitud en PostgreSQL Departamento de Informática, Universidad Nacional de San Luis, Ejército de los Andes 950, San Luis, Argentina. {fkasian, nreyes}@unsl.edu.ar}.

Kornblith, Miriam, Venezuela: calidad de las elecciones y calidad de la democracia. América.

Larrea, M. (2012) SOPORTES DE COMUNICACIÓN: EL SMS EN LAS CLASES DE LENGUA Teoría de la Educación. Educación y Cultura en la Sociedad de la Información, vol. 13, núm. 3, noviembre, 2012, pp. 176-189 Universidad de Salamanca, España (Disponible en: <http://www.redalyc.org/articulo.oa?id=201024652009>).

Latina Hoy [en línea] 2007, (abril): [Fecha de consulta: 3 de febrero de 2018] Disponible en: <http://www.redalyc.org/articulo.oa?id=30804506>, ISSN 1130-2887.

López, E. (2012) Estudio del consumo de energía en un dispositivo Android.

López, M. (2019) Venezuela 2019: nuevo intento por alcanzar la transición democrática (Disponible <https://margaritalopezmaya.com/>).

Machado, N. (2012). Diseño del proceso para la gestión de proyectos de sistemas de información. Caso de estudio: Dirección general de tecnología de la información del consejo nacional electoral. Universidad Católica Andrés Bello.

Mateu, C. (2004) Desarrollo de Aplicaciones web Software libre Fundació per la Universitat Oberta de Catalunya (<http://www.uoc.edu>).

Méndez y Heredia (2012) Evolución internacional y nacional de las Salas Situacionales en Salud. Revista Comunidad y Salud, Año 2012, Vol. 10(1), Ene-Jun, disponible en <http://servicio.bc.uc.edu.ve/fcs/cysv10n1/art07.pdf>.

Molina, J., Loja, N., Zea, M. y Loaiza, E. (2016) Evaluación de los Frameworks.

Murillo, J., y Caamaño, S. (2010) Vol24 - Num1 - Implementación de un servidor FTP utilizando el modelo cliente/servidor mediante el uso de sockets en lenguaje c UNIX con el fin de mejorar los tiempos de respuesta en la red.

Obando, A. y Ramírez, H. (2016) Diseño e implementación de un Sistema de Notificación y Control de Alarmas por medio del servicio SMS de la Red celular GSM para Dispositivos Móviles. Universidad Nacional de Ingeniería Recinto Universitario “Simón Bolívar” Facultad de Electrotecnia y Computación. Managua Nicaragua.

OBS Busines School (2016) Project Management Características y fases del modelo incremental Disponible:

<https://obsbusiness.school/int/blog-project-management/metodologias-agiles/caracteristicas-y-fases-del-modelo-incremental>).


Ochoa, E. y Villa, J. (2013). Sala situacional y procesos electorales en Venezuela:

Peñaloza, N. (2016) PROGRAMACIÓN ANDROID CON PYTHON Programación Paralela y Distribuida Facultad de Ingeniería UAEMex.

Primo, A. (2012) Protocolo HTTP World Wide Web consortium Oficina Española Teoría de SRI 12/01/2012.

Sagredo, J., Trueba, A., Martínez, M. y López M. (2012) Ciencia Ergo Sum, vol. 19, núm. 3, noviembre-febrero, 2012, pp. 239-250 Universidad Autónoma del Estado de México Toluca, México (Disponible en: <http://www.redalyc.org/articulo.oa?id=10423895005>).

Sartori, G. (2017) La Política democrática reivindicada (2017) Alianza Editorial, 1º edición “Ensayo”, España.

Villa, J., y Ochoa, E. (2013) Sala situacional y procesos electorales en Venezuela: Importancia del soporte técnico y estadístico. *Jurídicas CUC*, 9 (1), 37 – 61.

Zea, M., Molina, J., Redrován, F. (2017) ADMINISTRACIÓN DE BASES DE DATOS CON POSTGRESQL. Ingeniería y Tecnología 3 Ciencias Primera edición: abril 2017 Editorial Área de Innovación y Desarrollo, S.L (Disponible en <http://dx.doi.org/10.17993/IngyTec.2017.18>).

www.bdigital.ula.ve