



FACULTAD DE INGENIERÍA

Carrera de Ingeniería de Sistemas Computacionales

“SISTEMA WEB DE VOTO ELECTRÓNICO PARA EL PROCESO DE VOTACIONES DE ASAMBLEA GENERAL DE DELEGADOS EN LA COOPSERMUL 7 DE AGOSTO LTDA.”

Trabajo de suficiencia profesional para optar el título profesional de:

Ingeniero de Sistemas Computacionales

Autor:

Edgar Ismael Tasayco Altamirano

Asesor:

Mg. Franchesca Fiorella Rodríguez Rivera

Lima – Perú

2021

DEDICATORIA

En primer lugar, dedicárselo a Dios por darnos la salud y fortaleza necesaria de mantenernos con bien en momentos difíciles que atravesó el mundo con el COVID -19, en segundo lugar, a mi madre y mi padre Marcelino Tasayco Fajardo, quien se me fue en tiempos de pandemia, quien me formo con valores éticos y morales correctos, gracias padre, en tercer lugar, a mi esposa e hijo por su paciencia y su tiempo robado puesto en esta investigación, sin ustedes no lo podría haber logrado.

AGRADECIMIENTOS

Desde lo más profundo a Dios por darnos la salud, a nuestros padres por los valores formados que nos ayudan a emprender constantemente en seguir superándonos, a mi esposa a mi hijo, a todos aquellos miembros de nuestras familias que estuvieron en momentos cruciales y nos brindaron su apoyo incondicional, a nuestros profesores de la Universidad Privada del Norte que nos ofrecieron sus conocimientos para complementar nuestras vidas como profesionales de bien y sentir que disfrutamos nuestra profesión.

TABLA DE CONTENIDO

DEDICATORIA	ii
AGRADECIMIENTOS	iii
ÍNDICE DE FIGURAS.....	vii
ÍNDICE DE TABLAS	xi
ANEXOS.....	xii
1. CAPÍTULO I INTRODUCCIÓN	1
1.1. REALIDAD PROBLEMÁTICA.....	1
1.2. OBJETIVOS	11
1.3. JUSTIFICACIÓN	12
2. CAPÍTULO II MARCO TEÓRICO	14
2.1. ANTECEDENTES	14
2.2. PROCESO DE VOTACIÓN Y ASAMBLEAS	16
2.3. ARQUITECTURA GENERAL DEL SISTEMA WEB	24
2.3.1. <i>Metodología de desarrollo de software</i>	24
2.3.1.1. Proceso Unificado de Rational (RUP).....	25
2.3.2. <i>Spring Framework</i>	27
2.3.2.1. Módulos de spring.....	27
2.3.2.2. Inversión de control	29
2.3.2.3. Inyección de dependencia.....	32
2.3.2.4. Estereotipos	34
2.3.2.5. Anotaciones.....	35
2.3.3. <i>Aplicación web y herramientas</i>	38
2.3.3.1. Spring Tool Suite -versión 4.9.0 (Sts).....	38
2.3.3.1.1. Requisitos De Instalación Sts.....	38
2.3.3.2. Gestores de bases de datos.....	43

2.3.3.2.1.	Gestor de bases de datos relacional	43
2.3.3.2.2.	Gestores de bases de datos más utilizados	45
2.3.3.3.	Patrón de arquitectura	45
2.3.3.4.	Spring boot	47
2.3.3.4.1.	Maven	48
2.3.3.5.	Persistencia de datos.....	51
2.3.3.5.1.	Aplicaciones de persistencia Java (JPA).....	51
2.3.3.5.2.	Hibernate (ORM).....	52
2.3.3.5.3.	Relaciones de objetos	53
2.3.3.5.1.	Spring data Jpa	54
2.3.3.6.	Recursos de publicación	58
2.3.3.6.1.	Servidor privado virtual (VPS)	58
2.3.3.6.2.	Proveedores de VPS	58
2.3.3.6.3.	Servidor web Apache Tomcat	59
2.3.4.	Seguridad	60
2.3.4.1.	Spring security	61
2.3.4.2.	Security sockets layer (SSL)	69
2.3.4.3.	Criptografía.....	71
2.3.4.3.1.	Criptografía Asimétrica	71
2.4.	DEFINICIÓN DE TÉRMINOS BÁSICOS	72
3.	CAPITULO III DESARROLLO DEL PRODUCTO	75
3.1.	MODELO DE CASOS DE USO DEL NEGOCIO – MCUN (VISTA EXTERNA)	75
3.2.	MODELO DE ANÁLISIS DEL SISTEMA (MAN) – VISTA INTERNA.....	81
3.3.	MODELO DE CASOS DE USO DEL SISTEMA (MCUS).....	85
3.4.	ANÁLISIS Y DISEÑO DEL SISTEMA (MCUS)	92
4.	CAPÍTULO IV DESCRIPCIÓN DE LA EXPERIENCIA	98
5.	PANTALLAS DEL SISTEMA	99

5.1.	PRESENTACIÓN	99
5.2.	PANTALLAS QUE PUEDE VISUALIZAR EL ADMINISTRADOR DE LA PLATAFORMA	99
6.	FUENTES DEL SISTEMA.....	109
7.	RESULTADOS	115
8.	CONCLUSIONES.....	116
9.	BIBLIOGRAFÍA.....	117
10.	ANEXOS.....	119

Índice de Figuras

Figura1 Lugares de votación electrónica presencial - Onpe.....	3
Figura2 Api Web Onpe No Presencial.....	4
Figura 3 Proceso generación de asamblea (As- is).....	20
Figura 4 Proceso de votación actual	22
Figura 5 Componentes del proceso	26
Figura6 Módulos de Spring	28
Figura7 Acoplamiento de Clase - Ejemplo.....	31
Figura8 Inyección de Dependencia - ejemplo	33
Figura9 Estereotipos de Spring.....	34
Figura10 Sistema operativos para Spring Tool Suit	39
Figura11 Generar Nuevo Proyecto Spring Tool – Paso 1	39
Figura12 Generar Nuevo Proyecto Spring Tool – Paso 2	40
Figura13 Generar Nuevo Proyecto Spring Tool – Paso 3	41
Figura 14 Elegir Dependencias Paso 4	42
Figura 15 Patrón Modelo Vista Controlador MVC	46
Figura16 Esquema usando Maven.....	49
Figura17 Pom Xml que contiene las dependencias del proyecto	50
Figura18 Mapeo entre Tabla de base de datos y Objeto Java.....	53
Figura19 Consultas con Spring Data Jpa.....	56
Figura 20 Jpa Repository - Crud Repository	56
Figura 21 Inyección de dependencia con Spring.....	57
Figura22 Panel de control de Tomcat	60
Figura23 Configuración fail2ban - seguridad de consola.....	65

Figura24 Ip baneadas por intento de ataque	66
Figura 25 Seguridad a nivel de Url - Http request.....	68
Figura26 Seguridad a Nivel de Métodos	69
Figura27 Criptografía Asimétrica.....	72
Figura28 Estructura principal del MCUN	75
Figura 29 Diagrama de objetivos del negocio	76
Figura 30 Objetivos del negocio general - ON	76
Figura 31 Objetivo especifico.....	77
Figura 32 Casos de uso de negocio.....	78
Figura 33 Actores de negocio	79
Figura34 Diagrama general de casos de uso del negocio	80
Figura 35 Diagrama de objetivos del negocio vs casos de uso del negocio	80
Figura 36 Organizar modelo de análisis del negocio.....	81
Figura 37 Trabajadores de negocio.....	81
Figura 38 Realizaciones de negocio (RN)	83
Figura 39 Entidades de negocio (EN).....	83
Figura 40 Organizar MCUS.....	85
Figura 41 Actores del sistema.....	86
Figura 42 Paquetes de casos de uso del sistema	88
Figura 43 Casos de uso de Gestión de votaciones	89
Figura 44 Casos de uso de Seguridad del sistema	89
Figura 45 Diagrama general de CUS	90
Figura46 Casos de usos del sistema.....	91

Figura 47 Diagrama de clases	92
Figura 48 Organizar análisis y diseño.....	92
Figura 49 Análisis de gestión de votaciones.....	93
Figura 50 Análisis de seguridad del sistema.....	93
Figura 51 Diagrama de actividades – Gestionar Candidato	94
Figura 52 Diagrama de actividades – Gestionar Delegado.....	94
Figura 53 Diagrama de actividades – Gestionar Vacante.....	95
Figura 54 Diagrama de actividades – Gestionar Tema.....	95
Figura 55 Diagrama de actividades – Gestionar Voto.....	96
Figura 56 Diagrama de actividades – Gestionar Asistencia	96
Figura 57 Esquema de base de datos	97
Figura 58 Pantalla Login de usuario	99
Figura 59 Pantalla Menú del administrador.....	100
Figura 60 Pantalla Tipos de votaciones	100
Figura 61 Pantalla solo voto por Tema.....	101
Figura 62 Pantalla Selección de la Opción a votar por Tema.....	101
Figura 63 Pantalla Constancia de voto por Tema	102
Figura 64 Pantalla No voto Duplicado	102
Figura 65 Pantalla Selección de voto por Candidato.....	103
Figura 66 Pantalla Votar por su Candidato.....	103
Figura67 Pantalla Gestionar Delegado	104
Figura68 Pantalla Gestionar Tema	104
Figura69 Pantalla Gestionar Asistencia.....	105

Figura70 Pantalla Gestionar Candidato	105
Figura71 Pantalla Gestionar Vacante	106
Figura72 Pantalla Resultado de Tema	106
Figura73 Pantalla Resultado de Candidato - Voto Secreto.....	107
Figura74 Pantalla de Firma Digital - Criptografía asimétrica	108
Figura75 Pantalla Cambio de Contraseña.....	109
Figura 76 Fuente Árbol de paquetes	109
Figura 77 Fuente Pom Xml - Referencia maven	110
Figura 78 Fuente Aplicando Hibernate.....	111
Figura 79 Fuente Aplicando Spring Data	111
Figura 80 Fuente Aplicando Seguridad a nivel de Url	112
Figura 81 Fuente Seteando Clave privada - corografía asimétrica.....	112
Figura82 Fuente Seguridad a nivel de roles.....	113
Figura83 Fuente Plantilla de menú principal - Template thymeleaf	114

Índice de Tablas

Tabla 1 Gastos por Asamblea Presencial año 2019	6
Tabla 2 Tiempo promedio por votación presencial - voto nominal	7
Tabla 3 Tiempo invertido en votación presencial - voto nominal	8
Tabla 4 Tiempo estimado por Ronda de Votación Presencial - Voto secreto	9
Tabla 5 Tiempo estimado total por Voto Secreto Presencial.....	9
Tabla 6 Tiempo Total invertido en Votación.....	10
Tabla 7 Resumen de tiempo invertido en votación presencial.....	11
Tabla 8 Fases Del Proceso Unificado Racional	25

ANEXOS

Anexo 1 Publicación de convocatoria	119
Anexo 2 Ley 31029 - para voto electrónico en Cooperativas	120
Anexo 3 Rastros de firma digital	120
Anexo 4 Resultado de votación - para el Comité electoral	121
Anexo 5 Resultado de votación - para el Comité de vigilancia.....	121
Anexo 6 Resultado de votación - para el Consejo de administración	122
Anexo 7 Resultado voto por tema - Oportunidad de voto	122
Anexo 8 Resultado de votación - aprobación de dietas	123
Anexo 9 Reunión virtual con delegados	124

CAPÍTULO I. INTRODUCCIÓN

1.1. Realidad Problemática

El voto electrónico está presente en diversas partes de mundo, el impacto que puede generar su utilización, está sujeto a las herramientas tecnológicas que se emplean para su correcto funcionamiento que garanticen la seguridad, transparencia y el no repudio de las votaciones.

En Europa existen algunos países que utilizan el voto electrónico para sus diversos comicios electorales de manera sectorizada: “En diversos países de Europa se ha desarrollado diversos ensayos de voto electrónico en diferentes modalidades, también en partes de los estados de U.S.A, países de América central y del sur” (Panizo, 2010, pág. 7), “fue en la ciudad del Shelton de los U.S.A la primera experiencia de voto por internet, donde 103 votantes realizaron su voto electrónico desde su hogar” (Sanchez, 2016, pág. 67).

En diversos municipios de Holanda en el año 1995 se realizó el voto electrónico con resultados positivos, en la ciudad de Estrasburgo - Francia se utilizó el voto electrónico en elecciones presidenciales de mayo de 1995 en Issy-Les-Moulineaux y en elecciones europeas en junio de 1994 donde participaron aproximadamente 4000 electores, también en los países de Noruega y Dinamarca de realizaron diversas pruebas entre los años de 1992 y 1995 (Fernandez, 2001, pág. 207).

Existe un gran temor de que el sistema informático falle, que las urnas sean manipuladas, o que eliminen votos de manera accidental y se generen cambios en los resultados y se genere una mayor desconfianza entre los partidos políticos y la ciudadanía, la falta de conocimiento de las nuevas tecnologías para votantes de avanzada edad o población en extrema pobreza que le impidan contar con dispositivos electrónicos de votación (Gálvez & Gonzáles, 2011, pág. 261)

La seguridad informática es uno de los requisitos indispensables para brindar un buen servicio, sin embargo, esta tiene que ir acompañada de una buena arquitectura tanto de software

como se hardware que permitan establecer medidas optimas de seguridad y una rápida respuesta ante situaciones adversas a lo planificado por lo que su implantación podría resultar de alto costo.

El mayor problema con las máquinas fue que el lector óptico de algunas no pudo registrar ciertos tipos de tarjetones porque se habían humedecido y aumentado de grosor. La dificultad quedó solucionada ajustando las máquinas y reiniciando la elección, o bien, realizando el conteo manual de los votos, como se hacía en la modalidad convencional (Velasquez, 2012, pág. 168).

Existen 2 modalidades de voto electrónico, el voto electrónico presencial (VEP) y el voto electrónico no presencial (VENP), el voto VEP, básicamente es una urna electrónica ubicada en el lugar de votación donde los electores asisten para realizar su votación. El VENP es realizar su votación desde un lugar ajeno al centro de votación, generalmente al votante se le brinda sus accesos personales para que pueda conectarse mediante algún dispositivo móvil o aplicación web para realizar su votación.

Se basa en una aplicación WEB, que puede ser utilizada por el elector desde cualquier dispositivo conectado a internet en cualquier lugar del mundo. Todas las transacciones que se realizan, incluyen comunicaciones seguras (al estilo de las comunicaciones bancarias), con utilización de protocolos de encriptación. El servidor donde se aloja la aplicación también posee seguridades, para inhibir accesos dañinos. Una de sus ventajas es que tiende a maximizar y simplificar la participación de los electores (Pesado, 2015, pág. 41).

Según Onpe (2016), En el Perú el voto electrónico en sus diferentes modalidades (presencial y no presencial) se encuentra autorizado mediante la ley 28581 publicada el 25 de julio del año 2005, donde autoriza a la Oficina Nacional de Procesos Electorales (ONPE), de realizar dicha implementación de manera gradual, debiendo tomar en cuenta seguridad, flexibilidad, tecnología y

todas aquellas regulaciones que garanticen unas elecciones seguras y transparentes, es por ello en el año 2006 la ONPE mediante la ley 29603 emite las normas legales que contendrán dicha regulación.

En Perú desde el año 2011 se han venido realizado el voto electrónico presencial (VEP) en diferentes distritos y provincias del Perú tal como se muestra en la siguiente tabla.

Figura 1

Lugares de votación electrónica presencial - Onpe

FECHA	ELECCIÓN	DISTRITOS
9/12/2018	Referendúm Nacional 2018 / Segunda Elección Regional 2018	Rocchacc, El Porvenir, Los Chankas, Chaca, Pucacolpa La Punta, Mi Perú, Kintiarina, Roble, Pichos, Santiago de Túcuma, Pucayu, Castillo Grande, Santo Domingo de Anda, Pacarán, Chaclacayo, Cieneguilla, Santa Anita, Miraflores, San Borja, San Luis, Surquillo, Lince, Magdalena del Mar, Pueblo Libre, San Isidro, Barranco, La Molina, Santiago de Surco, Lurin, Pachacamac, Pucusana, Punta Hermosa, Punta Negra, San Bartolo, Santa Maria del Mar, La Yarada los Palos, Neshuya y Alexander Von Humbolt.
7/10/2018	Elecciones Regionales y Municipales 2018	Rocchacc, El Porvenir, Los Chankas, Chaca, Pucacolpa La Punta, Mi Perú, Kintiarina, Roble, Pichos, Santiago de Túcuma, Pucayu, Castillo Grande, Santo Domingo de Anda, Pacarán, Chaclacayo, Cieneguilla, Santa Anita, Miraflores, San Borja, San Luis, Surquillo, Lince, Magdalena del Mar, Pueblo Libre, San Isidro, Barranco, La Molina, Santiago de Surco, Lurin, Pachacamac, Pucusana, Punta Hermosa, Punta Negra, San Bartolo, Santa Maria del Mar, La Yarada los Palos, Neshuya y Alexander Von Humbolt.
10/12/2017	Elecciones Municipales Diciembre 2017	Pucayu, Castillo Grande, Santo Domingo de Anda, Rocchacc, El Porvenir, Los Chankas, Roble, Pichos, Santiago de Túcuma, Chaca, Kintiarina y La Yarada los Palos.
12/03/2017	Elecciones Municipales 2017	Pucacolpa, Neshuya y Alexander Von Humbolt
10/04/2016	Segunda Elección Presidencial 2016	Barranco, Chaclacayo, Cieneguilla, La Punta, Lince, Lurín, Magdalena Del Mar, Mi Perú, Pacarán, Pachacamac, Pucusana, Pueblo Libre, Punta Hermosa, Punta Negra, San Bartolo, San Isidro, San Luis, Santa María Del Mar, Surquillo.
10/01/2016	Elecciones Generales 2016	Barranco, Chaclacayo, Cieneguilla, La Punta, Lince, Lurín, Magdalena Del Mar, Mi Perú, Pacarán, Pachacamac, Pucusana, Pueblo Libre, Punta Hermosa, Punta Negra, San Bartolo, San Isidro, San Luis, Santa María Del Mar, Surquillo.
29/11/2015	Elecciones Municipales 2015	Mi Perú - Callao
7/12/2014	Segunda Elección Regional 2014	Pacarán
5/10/2014	Elecciones Regionales y Municipales 2014	La Punta, Pacarán, Santa María del Mar, San Bartolo, Pucusana, Punta Hermosa, Punta Negra
24/11/2013	Nuevas Elecciones Municipales para la Municipalidad Metropolitana de Lima 2013	Santa María del Mar
7/07/2013	Nueva Elecciones Municipales 2013	Pacarán
30/09/2012	Consulta Popular de Revocatoria 2012	Pacarán
5/06/2011	Elecciones Generales Segunda Elección Presidencial 2011	Pacarán

Fuente : (Onpe, 2021)

En la tabla nro. x se puede visualizar 3 columnas, la columna fecha hace referencia a las fechas en que se realizaron las votaciones electrónicas presenciales, la columna de elección, muestra

el nombre que recibe el cómicos electoral y la columna de distritos, hace referencia en los distritos en que se han realizado las votaciones electrónicas,

Estos procesos de voto electrónico presenciales realizados por la ONPE fueron auditados por organismos internacionales que corroboran el cumplimiento de estándares solicitados para este tipo de comicios electrónico.

La ONPE también cuenta con el voto electrónico no presencial (VENP), es un sistema realizado para el voto electrónico a distancia por internet que ha tenido una mayor utilización en el año 2020 debido a las medidas de distanciamiento social que atraviesa el mundo, utilizado por empresas e instituciones del sector público y privado.

Figura2

Api Web Onpe No Presencial



Práctica de Votación
Utilice el teclado virtual para ingresar sus credenciales

1 Número de PIN de 10 dígitos
Para practicar coloque los siguientes números: 1000-0000-20

2 Contraseña
Para practicar coloque los siguientes números: 30241283

3 Clave de la imagen
Para practicar coloque los siguientes números: 113829

DIRECTIVOS DEL COMITÉ EJECUTIVO NACIONAL		CONSEJO REGIONAL	
Haga clic sobre el candidato o número de lista de su preferencia		Haga clic sobre el candidato o número de lista de su preferencia	
DR. NOMBRES Y APELLIDOS A	55	DR. NOMBRES Y APELLIDOS E	59
DR. NOMBRES Y APELLIDOS B	56	DR. NOMBRES Y APELLIDOS F	60
DR. NOMBRES Y APELLIDOS C	57	DR. NOMBRES Y APELLIDOS G	61
DR. NOMBRES Y APELLIDOS D	58	DR. NOMBRES Y APELLIDOS H	62
VOTO NULO	VOTO NULO	DR. NOMBRES Y APELLIDOS I	63
		DR. NOMBRES Y APELLIDOS J	64
		VOTO NULO	VOTO NULO

SIGUIENTE

Fuente : (Onpe, 2021)

La figura muestra la aplicación web que tiene la Onpe para realizar votaciones a distancia por internet, donde inicialmente nos pide que ingresemos algunos caracteres de validación, luego de completar muestra una segunda pantalla donde se tiene que elegir la opción que tiene que marcar, dar clic en siguiente y aparecerá una tercera ventana para que pueda emitir su votación.

Las Cooperativas es uno de los rubros de cada país que también realizan diferentes tipos de votaciones internas normado por sus estatutos o reglamentos internos, también contemplan dos tipos de asambleas; la asamblea ordinaria y la asamblea extraordinaria. Dentro de estas asambleas se realizan una serie de votaciones para la aprobación o desaprobación de los diversos puntos de agenda.

La Cooperativa de Servicios Múltiples de Agosto Ltda., cada año realiza su asamblea general ordinaria de delegados provenientes de diferentes partes del país, para ser exactos son: sesenta y nueve (69) delegados de la capital de Lima y quince (15) delegados de los diferentes departamentos del Perú. Cada delegado es la representación de un grupo de socios y son el nexo de coordinación para asistirlos a cualquier trámite dentro de la Cooperativa. A la asamblea también asisten los directivos de la Cooperativa, siete (7) directivos del consejo de administración, tres (3) directivos del consejo de vigilancia y tres (3) directivos del comité de electoral, además de una (1) persona a cargo de la filmación, una (1) persona de soporte técnico, una (1) personal a cargo de la limpieza y diez (10) trabajadores entre seguridad, vigilancia y administrativos de apoyo. Las asistencias de estas personas a la asamblea generan un costo que deben ser cubierto por la Cooperativa, para ello se muestra a continuación el detalle de estos gastos.

Tabla 1

Gastos por Asamblea Presencial año 2019

Descripción	Monto	Cantidad	Nro. Días	Total
Desayuno	12.00	110	1	1,320.00
Almuerzo	25.00	110	1	2,750.00
Cena	25.00	110	1	2,750.00
Hospedaje	40.00	15	2	1,200.00
Pasajes	320.00	15	1	4,800.00
Filmación	1,500.00	1	1	1,500.00
Otros	3,500.00	1	1	3,500.00
TOTAL				17,820.00

Fuente: Cooperativa de Servicios Múltiples 7 de agosto Ltda.

En la Tabla N.º 1 se puede apreciar los diferentes gastos que involucra la realización de la asamblea, para el concepto de gasto por Desayuno para todos los asistentes a la asamblea es de S/. 1,320.00 nuevos soles, para el Almuerzo y la Cena es de S/. 2,750.00 nuevos soles cada concepto, para el Hospedaje de los Delegados que vienen de provincia es de S/. 1,200.00 nuevos soles que cubre el costo del hospedaje por dos días, para el concepto de Pasajes se está promediando el monto ya que el pasaje de cada provincia puede variar de acuerdo a la lejanía es por ello que el monto es de S/. 4,800.00 nuevos soles que cubre el pasaje de ida y retorno, para el concepto de Filmación es de S/. 1,500.00 nuevos soles, también existe el concepto de Otros que involucra gastos diversos y se están promediando por el importe de S/. 3,500.00 nuevos soles, este concepto cubre los gastos por el pago por seguridad, apoyo administrativo, bebidas, aguas, menajes, etc. La suma total de este gasto es de S/. 17,820.00 nuevos soles y son en su totalidad cubierto por la Cooperativa.

En el año 2019 la Asamblea General de Delegados se inició a las 09:00 am y termino a las 21:45 pm, el tiempo de duración va a depender de la cantidad de puntos o temas tratados en la asamblea, existen dos escenarios de temas diferentes a cargo dos autoridades de la Cooperativa.

1. El primer escenario está a cargo del Consejo de Administración, en ese espacio se otorga informes, pedidos y acuerdos de temas relacionados a la Cooperativa. Todos estos puntos son llevados a votación simple y para realizar este procedimiento realizan de la siguiente manera, el presidente lee el tema a llevar a votación, un representante del consejo de vigilancia es el encargado de llevar el conteo de las respuestas de cada delegado para ello el representante del consejo de vigilancia pregunta a cada delegado si su respuesta es a favor, en contra o se abstiene. Para reflejar el tiempo que puede tardar cada ronda de votación se muestra la siguiente tabla.

Tabla 2

Tiempo promedio por votación presencial - voto nominal

Cantidad de votaciones	Nro. de participantes	Tiempo por votación	Tiempo en segundos	Tiempo en minutos
1	84.00	15 segundos	$84 \times 15 = 1260$ s	21 min

Fuente: Cooperativa de Servicios Múltiples 7 de agosto Ltda.

En la Tabla Nro.2 se establece el tiempo promedio que puede durar cada ronda votación simple donde participan los 84 delegados que asistieron a la asamblea y tiene una duración de 21 minutos.

Además, en este tipo de asambleas no se tiene precisado en número de votaciones que puedan existir ya que eso va a depender de los diferentes temas que se toquen y lo lleven a votación. En el año 2019 se realizaron 10 rondas de votación y para conocer el tiempo total invertido en realizar la votación se muestra la siguiente tabla.

Tabla 3

Tiempo invertido en votación presencial - voto nominal

Cantidad de votaciones	Nro. de participantes	Tiempo por votación	Tiempo en minutos	Tiempo en horas y minutos
10	84.00	21 minutos	10 X 21 =210 min	3 hr y 30 min

Fuente: Cooperativa de Servicios Múltiples 7 de agosto Ltda.

En la Tabla Nro. 3 se muestra que en el año 2019 se realizó 10 rondas de votación simple donde participaron los 84 delegados y en referencia a la Tabla Nro. 1 donde el tiempo por cada ronda es de 21 minutos, este tiempo es multiplicado por las 10 rondas de votación dándonos un tiempo de 3 horas y 30 minutos.

2. El segundo escenario está a cargo del Comité Electoral, quienes tiene la facultad de elegir a los nuevos directivos mediante el voto secreto, para realizar este procedimiento el Comité de Electoral registra a los candidatos a quienes le asignan un número que los identifica, estos candidatos son propuestos por los delegados de la asamblea, luego adecuan una cabina de votación para que cada delegado se acerque y registre el número de su candidato y los inserte a la urna de votación,

terminado de emitir el total de delegados su votación, se procede al conteo y lectura de los resultados, para reflejar el tiempo invertido en este proceso de voto secreto se muestra la tabla a continuación.

Tabla 4

Tiempo estimado por Ronda de Votación Presencial - Voto secreto

Cantidad de votaciones	Nro. de participantes	Tiempo por voto por participante	Tiempo en minutos	Tiempo por conteo de resultados	Tiempo en horas y minutos
1	84.00	1 minuto	84 x1 = 84	20 minutos	1 hr y 44 min

Fuente: Cooperativa de Servicios Múltiples 7 de agosto Ltda.

Tabla 5

Tiempo estimado total por Voto Secreto Presencial

Cantidad de votaciones	Nro. de participantes	Tiempo por voto por participante	Tiempo en minutos	Tiempo por conteo de resultados
1	84.00	1 minuto	84 x1 = 84	20 minutos

Fuente: Cooperativa de Servicios Múltiples 7 de agosto Ltda.

En la Tabla Nro. 5 se puede apreciar el tiempo para realizar una ronda de voto secreto donde participan los 84 delegados, cada delegado tiene 1 minuto para ejercer su voto secreto, el resultado

del tiempo es de 84 minutos, a ello se le suma el tiempo por conteo de resultados que es de 20 minutos dando un tiempo total de 1 hora y 44 minutos (Tabla 4).

La cantidad de ronda de voto secreto va a depender de las vacantes disponibles para cada área de la Cooperativa, es por ello que solo en el año 2019 se realizaron 3 rondas de votación para ocupar las vacantes para el Consejo de Administración, Consejo de Vigilancia y Comité Electoral. Para reflejar el tiempo de estas tres rondas de votación de muestra la siguiente tabla.

Tabla 6

Tiempo Total invertido en Votación

Cant. de votaciones	Nro. de participantes	Tiempo por votación	Calculo de tiempo	Tiempo en horas y minutos
3	84.00	1 h 44 minutos	(1 h 44 min. * 3)	5 h 12 min.

Fuente: Cooperativa de Servicios Múltiples 7 de agosto Ltda.

En la Tabla Nro. 6 se puede apreciar el tiempo para realizar 3 rondas de votación secreta para los 84 delegados y en referencia a la Tabla Nro. 4 donde nos dice que por cada ronda el tiempo es de 1 hora con 44 minutos y multiplicándolo por las 3 rondas dando un tiempo total de 5 horas con 12 minutos.

Para obtener el total de tiempo invertido en los dos escenarios de la asamblea, es decir el tiempo total para el voto simple y el tiempo total para el voto secreto, se muestra la siguiente tabla.

Tabla 7

Resumen de tiempo invertido en votación presencial

Concepto	Rondas de votaciones	Tiempo
Voto simple	10	3 h 30 min
Voto secreto	3	5 h 12 min
TOTAL	13	8 h 42 min

Fuente: Cooperativa de Servicios Múltiples 7 de agosto Ltda.

En la Tabla Nro. 7 se muestra el total de tiempo por cada tipo de voto en el año 2019, para el voto simple en 10 ronda de votaciones tiene un tiempo total de 3 horas con 30 minutos, para el voto secreto en 3 ronda de votaciones tiene un tiempo total de 5 horas con 12 minutos. El tiempo total para los dos tipos de voto es de 8 horas con 42 minutos.

Los fenómenos identificados en esta investigación están ligados a un tipo de solución que permita realizar la participación de los delegados a la asamblea general ordinaria y puedan ejercer su votación desde cualquier dispositivo (Pc, tableta o celulares) de manera óptima y que soporte cualquier sistema operativo, para ello se necesitaría una plataforma de voto electrónico web flexible, transparente y segura.

1.2. Objetivos

Objetivo general

Implementar un sistema web de voto electrónico para optimizar el proceso de votaciones de asamblea en la Coopsermul “7 de agosto” Ltda.

Objetivos específicos

- Desarrollar el modelado del proceso del negocio
- Desarrollar la captura de requisitos de sistema
- Desarrollar el análisis y diseño del sistema
- Desarrollar la implementación del sistema
- Desarrollar las pruebas del sistema
- Desarrollar el despliegue del sistema

1.3. Justificación

Justificación teórica

En la presente tesis servirá como base para otros investigadores que quieran implementar un sistema de voto virtual utilizando herramientas tecnológicas, y de base de apoyo para aquellos investigadores que quieran implementar cualquier otro sistema informático y quieran cumplir con todas las etapas que implica la construcción de un software, en este proyecto de está utilizando la metodología RUP (Rational Unified Process) de apoyo para el análisis, implementación y documentación del proyecto. También se utiliza el Spring framework como marco de trabajo, Se utilizan los estereotipos de Spring para crear un patrón de arquitectura más flexible basado en el patrón de arquitectura tradicional MVC (Modelo, Vista, Controlador), Spring boot como herramienta de creación del proyecto que nos simplificara enormemente las configuraciones de dependencias para solo céntranos en la programación, Spring security para la seguridad de nuestra aplicación, Hibernate y Spring data para la persistencia y simplificación de creación de métodos que nos permitirá trabajar con cualquier base de datos. Utilizando las diferentes tecnologías basadas en Spring permitirá crear un software flexible y robusto de alto nivel.

Justificación práctica

La presente investigación se justifica en la práctica porque permitirá reducirá el tiempo de duración de la asamblea general de delegados y, además:

- Se podrán realizar más rondas de votaciones en un menor tiempo.
- Se reducirán los costos por organización de asambleas físicas.
- Contemplará el escenario del voto simple y voto secreto.
- Respetar las leyes normadas por el estado respecto al distanciamiento social.

Justificación social

De acuerdo al Decreto Supremo N° 008-2020-SA, publicado en el diario el Peruano el 11 de marzo del 2020 en su artículo 2 Medidas de prevención y control para evitar la propagación del COVID-19 en su punto 2.1.3 Espacios públicos y privados dice : a) En el caso de actividades o eventos que impliquen la concentración de personas en espacios cerrados o abiertos que ofrezcan mayores riesgos para la transmisibilidad del COVID-19, corresponde a la autoridad competente evaluar los riesgos para determinar la pertinencia de su realización.

•Da cumplimiento a la leyes de acuerdo a lo siguiente, Con fecha 14 de julio del 2020, se publica en el diario el peruano la Ley 31029 lo siguiente: Facúltese a los consejos y comités de las organizaciones cooperativas, constituidas al amparo del Decreto Supremo 074-90-TR, Texto Único Ordenado del Decreto Legislativo 085, Ley General de Cooperativas, la realización de sesiones no presenciales o virtuales, a través del uso de medios tecnológicos o telemáticos y de comunicaciones que permitan la participación, la comunicación, el ejercicio del voto y garanticen la autenticidad de los acuerdos que se adopten; los requisitos para su convocatoria y adopción de acuerdos serán establecidos en el estatuto de la cooperativa.

CAPÍTULO II. MARCO TEÓRICO

2.1. Antecedentes

Nacionales

1. En el año 2016 (Zavaleta) en su tesis de doctorado de nombre “VOTO ELECTRÓNICO EN LIMA METROPOLITANA, 2016”, la misma que tiene por objetivo proponer un sistema para la aceptación del voto electrónico en Lima Metropolitana. Para el desarrollo de su investigación utilizó instrumentos científicos tales como la encuesta y las entrevistas a electores que participaron del voto electrónico.

La autora concluye que el voto electrónico es una modalidad confiable y rápida teniendo las herramientas tecnológicas apropiadas que permitan brindar protección ante un fallo o caída del sistema, pero que existe aún desconfianza por parte de los electores por falta de información respecto al voto electrónico.

La investigación citada se vincula a la planteada, ya que una mala arquitectura tecnológica podría impactar de manera negativa ante la aceptación de los electores.

2. En el año 2013 Aradiel Castañeda Hilario en su tesis de grado de nombre “SISTEMATIZACIÓN DEL VOTO ELECTRÓNICO DE LA UNIVERSIDAD NACIONAL DEL CALLAO”, resume lo siguiente:

El presente estudio tuvo como objetivo fundamental sistematizar el voto electrónico presencial de la Universidad del Callao, indica que el proceso de votación será mucho más rápido, el votante podrá ser identificado rápidamente para generar su votación y los resultados serán develados en un menor tiempo, además indica que con dicha información almacenada se podrá

generar el informe de manera rápida, se podrá detectar posibles a personas que quieran realizar el voto ajeno además de ahorro en costos de personal que se realiza en una votación presencial.

3. Moroco (2010) en su tesis de nombre “ARQUITECTURA DE SISTEMAS DE SEGURIDAD PARA LA VOTACIÓN ELECTRÓNICA REMOTA”, resume lo siguiente: Que realizar la votación por internet es la más idónea para reemplazar al voto presencial, indica que, planteando un buen mecanismo de seguridad basado en cifrado de las votaciones y develarlos sin que estos sean visualizados, garantiza transparencia del proceso, además de tener un buen mecanismo de identificación basado en tarjetas inteligentes, ayudaran a eliminar la suplantación de votantes, el autor sugiere la utilización de los mecanismos mencionados para asegurar un correcto proceso electoral, tomando como referencia los procesos electorales que se realizaban los países de Francia, EEUU y España. Además, indica que utilizar este mecanismo será fácilmente de poder ser auditado.

Internacionales

1. Schmidt-Peralta y Gutiérrez-Alfaro (2015) en su proyecto de investigación titulada “HACIA EL DESARROLLO DE UN PROTOTIPO DE SISTEMA DE VOTO ELECTRÓNICO PARA EL CONTEXTO COSTARRICENSE”, desarrollada en Instituto Tecnológico de Costa Rica, donde participaron 147 estudiantes, donde 86 estudiantes votaron de manera electrónica y 61 estudiantes en la mesa tradicional de votación, para conocer su experiencia de votación realizaron encuestas donde el 98 % de estudiantes que usaron el voto electrónico indicaron que el sistema es muy fácil de utilizar

y solo el 2 % indico que le parecio dificil, dando como resultados de aceptacion muy aceptable en la modalidad electrónica.

2. Vaca et al. (2019) en su proyecto de tesis de nombre: “DESARROLLO DE UN SISTEMA PILOTO PARA EL EJERCICIO DEL SUFRAGIO EN LOS PROCESOS ELECTORALES INTERNOS EN INSTANCIAS PUBLICAS O PRIVADAS A TRAVÉS DE MEDIO A VIRTUALES (INTERNET) PARA EL CNE, resume lo siguiente: Utilizar una correcta arquitectura de software basado en el patrón de arquitectura Modelo Vista Controlados en combinación con algún framework de persistencia y la utilización de certificados digitales que mediante su algoritmo de encriptación que se encarguen de cifrar el voto del elector, cumpliría con los requerimientos solicitado por el Consejo Nacional Electoral.

3. Monserrate (2021) en su trabajo de investigación titulada “EL VOTO ELECTRÓNICO ALTERNATIVA PARA EL PROCESO ELECTORAL ECUATORIANO EN TIEMPOS DE PANDEMIA” resume lo siguiente: que es necesario contar con leyes adecuadas que permitan hacer que el voto electrónico tenga un mayor protagonismo en tiempos de pandemia para que el votante pueda realizar su votación sin tener que poner en riesgo su salud al tener que asistir a lugares públicos concurridos.

2.2. Proceso de votación y asambleas

El proceso de votación dentro de la Coopsermul es una obligación que se tiene que realizar al menos dos veces al año para dos tipos de escenarios totalmente diferente y en diferentes tiempos. Un tipo de votación se realiza en las asambleas generales donde solo participan delegados y

directivos para la renovación de nuevos directivos y aprobación de temas relacionados al negocio de la Coopsermul y el otro es para la elección de delegados, donde participan el total socios a nivel nacional para la elección de su delegado representante.

Requisitos para participar en asamblea general

Para que los delegados puedan participar en la asamblea general tienen que estar al día en sus aportaciones como socio, no tener deuda de préstamos con la empresa y no tener sanciones administrativas, esta verificación está a cargo del Comité Electoral. Este procedimiento se realiza con la finalidad de que el delegado se encuentre apto para poder postular a algún cargo administrativo durante la asamblea

Asamblea general ordinaria

La Asamblea General Ordinaria se celebra una vez al año obligatoriamente dentro de los 90 días calendarios posteriores a la terminación del ejercicio económico anual, bajo responsabilidad del Consejo de Administración. Algunos de los temas que se tienen que resolver en este tipo de asamblea se muestran a continuación

- a) Examinar los estados financieros y aprobar el balance general del ejercicio anual.
- b) Aprobar el Presupuesto Anual de la Cooperativa.
- c) Aprobar el Presupuesto del Consejo de Administración, la distribución de remanentes y excedentes.
- d) Elegir y remover por causas justificadas a los miembros de los Consejos y del Comité Electoral.
- e) Imponer sanciones de suspensión o destitución al cargo de directivos o delegados.

- f) Adoptar, en general, acuerdos sobre asuntos de importancia que afecten el interés de la Cooperativa.

Asamblea general extraordinaria

Este tipo de asamblea puede celebrarse en cualquier momento, cuando los intereses de la Cooperativa lo requieran, algunos de los temas podrían ser:

- a) Reformas de Estatuto.
- b) Resolver sobre la fusión, disolución y liquidación de la Cooperativa.
- c) Aprobar la transformación de la Cooperativa en un rubro diferente.

Los puntos mencionados en los dos tipos de asamblea son llevados a votación simple para su aprobación o denegación del tema.

Tipos de votación

Existen dos tipos de votos que se realizan en la asamblea general ordinaria, solo los delegados que hayan pasado su asistencia puede participar y estos tipos son:

- **Voto por mayoría simple**

Es el tipo de voto es realizado para la aprobación o denegación de temas que se y tienen las tres alternativas: A favor, en contra y la Abstención. Este tipo de voto no es secreto por lo tanto se puede emitir el listado de delegados que votaron a favor, en contra o abstención, siempre que la asamblea lo apruebe.

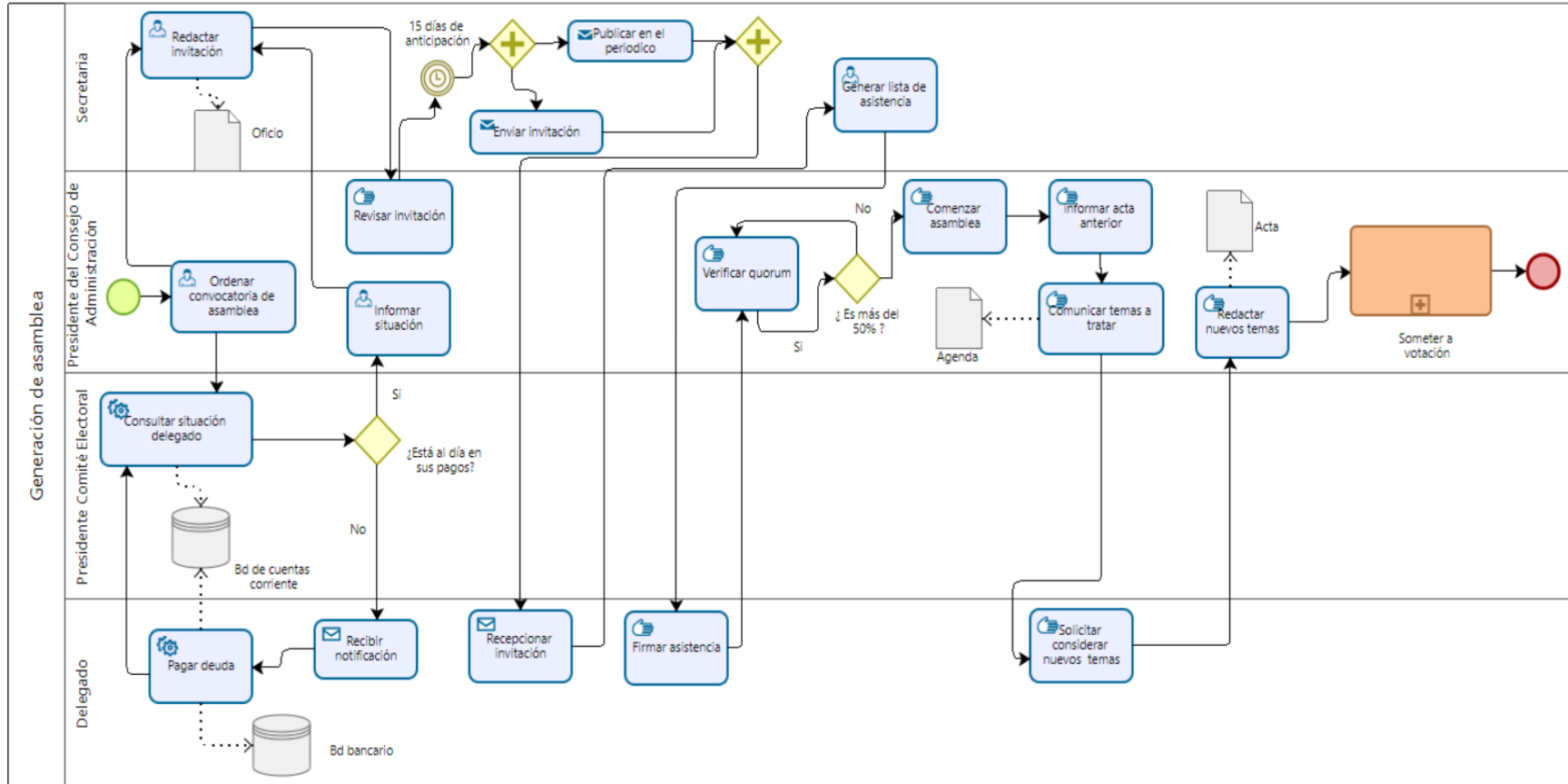
- **Voto secreto**

Es el tipo de voto para elección de candidatos, los delegados eligen al candidato de su preferencia y emiten su votación.

Reglas de votación en la asamblea

1. El delegado tiene que estar al día en su record de pagos y no tener sanciones administrativas.
2. La asamblea tiene que estar constituida por el 50 % +1 de asistencia de delegados.
3. El delegado tiene que estar presente en la asamblea y haber contestado presente.
4. Solo pueden votar delegados.
5. El delegado es libre de elegir la opción de por qué tema o porque candidato votar.
6. El delegado solo puede votar una sola vez por cada ronda de votación.
7. Las votaciones por voto simple a mano alzada no es secreto.
8. Las votaciones por voto secreto o por candidato, no se revela la identificación del votante.
9. El consejo de administración está a cargo del voto simple.
10. El comité electoral está a cargo del voto secreto.
11. El consejo de vigilancia actúa como ente supervisor y transparencia.

Figura 3
Proceso generación de asamblea (As- is)

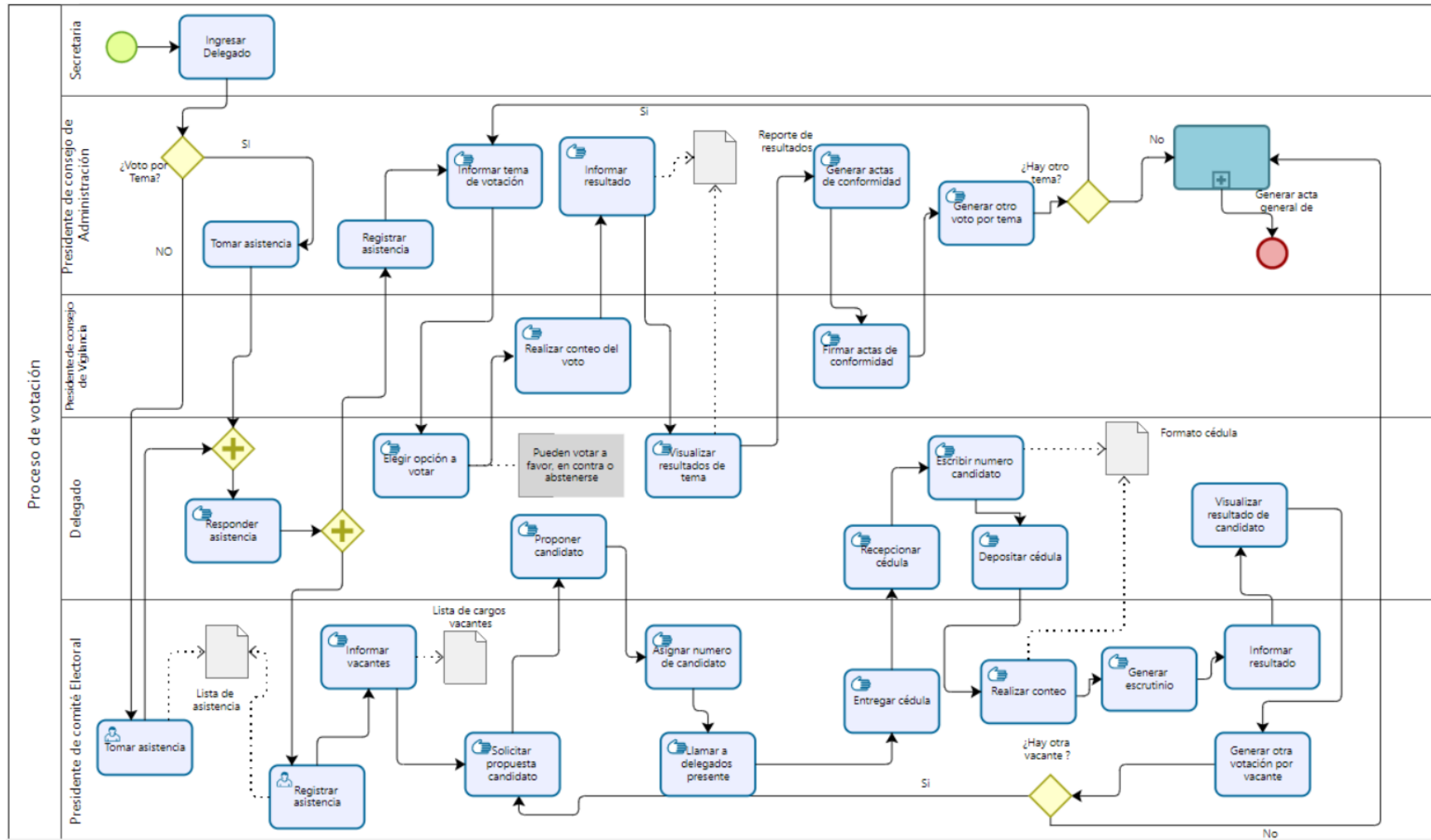


Fuente: Elaboración propia

En la figura 3 muestra el proceso actual para poder generar la asamblea general ordinaria de delegados en la Coopsermul 7 de agosto Ltda., en este proceso inicial intervienen cuatro (04) actores como es el presidente del consejo de administración, la secretaria, el presidente del comité de electoral y el delegado. Todo inicia cuando el presidente del consejo de administración ordena generar la convocatoria, el presidente del comité de electoral revisa la situación de todos los delegados que participaran en la asamblea con la finalidad de que todos se encuentren actos para ser candidato a algún cargo vacante en la asamblea, si un delegado no está al día en sus pagos, es notificado, el delegado realiza el pago por depósito a la cuenta bancaria de la empresa o puede acercarse a la empresa para que su pago sea registrado en el sistema de cuentas corrientes , el presidente del comité de electoral vuelve a revisar su situación para luego informar al presidente del consejo de administración la situación de todos los delegados, el presidente del consejo de administración ordena a la secretaria a redactar el oficio de invitación y publicación con 15 días de antelación, la secretaria envía la invitación a los delegado y al periódico para su difusión, el delegado recibe la invitación, el día de la asamblea la secretaria genera el listado de delegados que asistirán al recinto, el delegado firma la lista de asistencia, el presidente del consejo de administración verifica la lista firmada y si hubiera más del 50% de participantes da por establecida la asamblea si fuera menos del 50%, otorga un tiempo prudente para completar el quorum. Completado el quorum el presidente da lectura del acta de la asamblea anterior y comunica los puntos de agenda que se trataran, los delegados pueden sugerir incluir nuevos temas y el presidente añade los temas a tratar y los somete a votación

Figura 4

Proceso de votación actual



Fuente: Elaboración propia

En la figura 4 muestra el proceso para someter a votación, este proceso puede ser un voto simple o un voto secreto, ambos tipos de voto se realizan dentro de la asamblea sin embargo tienen un orden a seguir, inicialmente se realiza el voto simple que abarca todas las votaciones por aprobación o desaprobación de puntos de agendas y todo inicia cuando el presidente del consejo de administración informa a los delegados sobre el tema a votar y pregunta a los delegados que están a favor que levanten la mano, el presidente del consejo de vigilancia es el encargado de realizar el conteo, de la misma manera a los delegados que están en contra y los que se abstienen. Al finalizar el conteo de las tres opciones, a favor en contra y abstención, el presidente del consejo de vigilancia informa los resultados al presidente del consejo de administración quien informa los resultados, el delegado visualiza los resultados y el presidente del consejo de administración genera el acta y solicita a dos delegados representantes que firmen el acta en señal de conformidad, si hubiera otro tema de votación el proceso se repite sino de da por terminado el voto simple. Cuando el voto simple ha terminado inicia el voto secreto que básicamente es para elegir a los nuevos directivos, todo inicia cuando el presidente del comité de electoral toma asistencia, el delegado tiene que responder la asistencia para ser considerado presente, si no contesta la lista el delegado no será considerado para poder ejercer su voto, el presidente del comité electoral va actualizando su lista de asistencia y una vez terminado informa a los delegados los cargos vacantes y solicita a los delegados que emitan sus candidatos propuestos, el delegado propone a su candidato, el presidente del comité de electoral registra al candidato y asigna un número que lo representará a la hora que entrega las cédulas, solicita a dos delegados que respalden al candidato propuesto, los delegados respaldan la propuesta y el presidente de electoral vuelve a registrar a los delegados que respaldaron la propuesta, luego el presidente da inicio al voto secreto y va llamando a cada delegado uno a uno para entregarle su cédula e indica que digiten el número de su candidato en

dicha cedula, el delegado recibe la cedula y digita el número de su candidato de su preferencia, luego deposita la cedula en el ánfora de votación, una vez terminado de llamar a todos los delegados, el presidente del comité de electoral inicia el conteo de votos e informa los resultados de votación y llena el escrutinio, si hubiera otra vacante , el proceso se repite, sino lo hubiera se termina el proceso, una vez terminado el proceso de votación entrega los resultados al presidente del consejo de administración mediante un subproceso interno administrativo genera el acta general de la asamblea y termina el proceso de asamblea.

2.3. Arquitectura general del sistema web

La arquitectura general del sistema web está compuesta por todo aquello que interviene para su análisis, diseño y construcción del proyecto, sin dejar de lado la seguridad, la interfaz de usuario y herramientas de publicación en producción aplicando las mejores prácticas y tecnología innovadora.

2.3.1. Metodología de desarrollo de software

Una metodología de desarrollo de software es una secuencia de procedimientos que se deben aplicar en cada etapa de un proyecto, haciendo uso de diversas técnicas, métodos o modelos a seguir de manera eficiente y rigurosa. Existen diferentes tipos de metodologías que se pueden utilizar actualmente, la elección de alguna de ellas va a depender de la necesidad de cada proyecto sin embargo todas tienen como finalidad trabajar de la mano en la construcción de un software. (Maida & Pacienza, 2015).

2.3.1.1. *Proceso Unificado de Rational (RUP)*

RUP es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Asimismo, provee a través de un entorno WEB: Lineamientos, plantillas, workflows y herramientas, que guían una implementación efectiva de las mejores prácticas de la industria del software (Fernando Sánchez & Wong Portillo, 2010).

Rup trabaja bajo cuatro fases, en cada fase se tienen que desarrollar diferentes actividades, a continuación, se muestra la siguiente tabla:

Tabla 8
Fases Del Proceso Unificado Racional

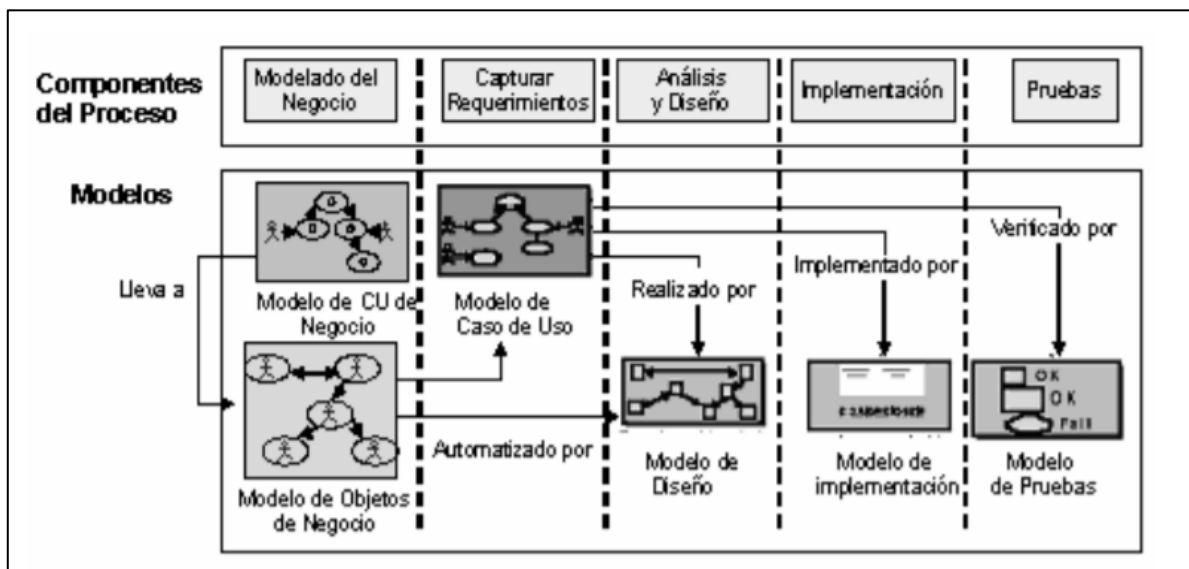
Flujo	Productos	Inicio	Elaboración	Construcción	Transición
Administración del Proyecto	Plan de desarrollo	I	R	R	R
	Caso de negocio	I	R		
	Lista de riesgos	I	R	R	R
Requisitos	Modelo de casos de uso	I	R		
	Visión	I	I	R	
	Especificación adicional	I	I	R	
	Glosario Modelo de diseño	I	I	R	R
	Modelo de diseño		I	R	

Análisis y Diseño	Documentación de la arquitectura SW		I		
Implementación	Plan de implementación		I	R	R
Test	Plan de test		I		
Despliegue	Plan de despliegue				I

Fuente: (Martínez & Martínez, 2014)

Nota. I = inicio, R = refinamiento

Figura 5
Componentes del proceso



Fuente (Garcia, 2006, pag. 8)

2.3.2. Spring Framework

Spring Framework es una ayuda para el desarrollo de aplicaciones corporativas en java. Surgió como una alternativa ligera a la compleja plataforma J2EE, y ha logrado una serie de popularidad entre los desarrolladores. Provee una serie de funcionalidades, de entre las cuales destacan la inyección de dependencias (ID), gestión de transacciones y soporte a pruebas automatizados. Spring Framework es un software libre, desarrollado por la Spring Source (Méndez Gonzáles, 2016).

Podemos decir que Spring Framework es un marco de trabajo código abierto basado en java que cuenta con funcionalidades estandarizadas basado en la automatización y reutilización de código que ha logrado tener una gran aceptación por los desarrolladores.

Spring Framework está basado en el principio de Inversión de Control (IoC), donde es el framework quien toma el control del código cambiando el paradigma de desarrollo donde el programador tiene el control del código. Además, aplica la Inyección de dependencias (DI) que es un patrón de desarrollo encargado de hacer la vinculación con otras clases por medio de interfaces. Cuenta con toda una gran familia de sub proyectos basado en buenas prácticas de desarrollo para la seguridad, persistencia, automatización, etc., que se integran en las diferentes capas del proyecto permitiendo tener un producto final robusto, seguro y flexible.

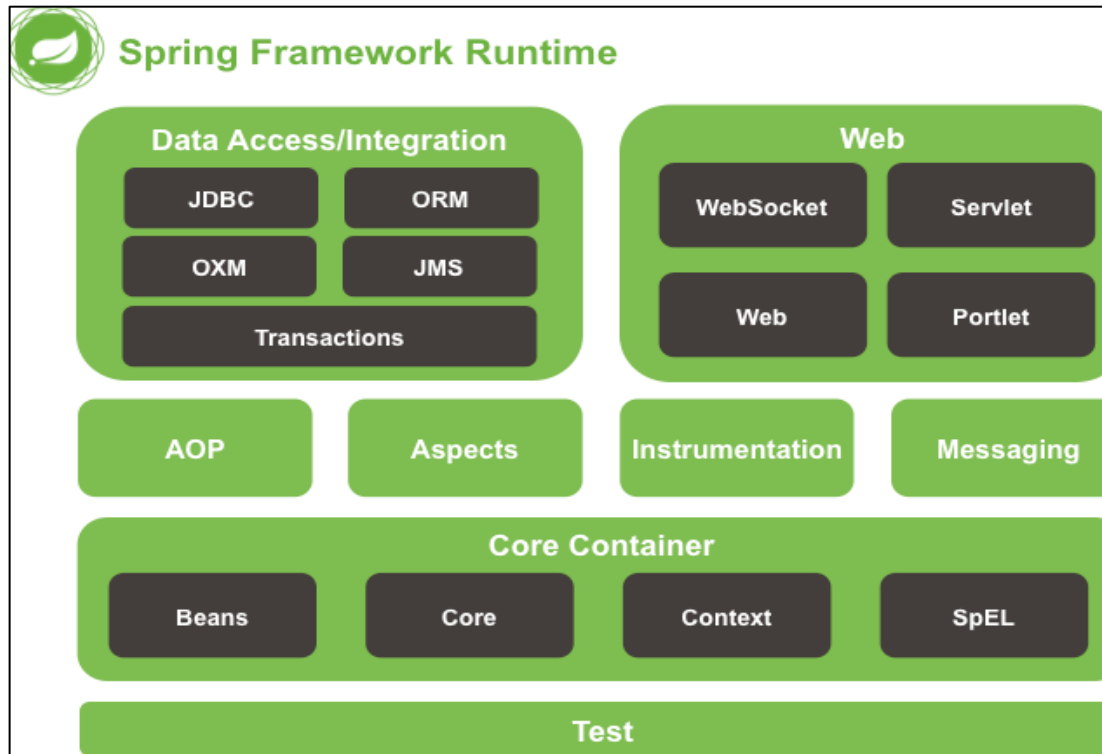
2.3.2.1. Módulos de spring

Spring Framework cuentan con diversos módulos para cada parte de la construcción de nuestra aplicación siendo el modulo más importante el Core Container, la elección de

los componentes que elijamos de Spring Framework va a depender de la necesidad del proyecto.

Figura6

Módulos de Spring



Fuente: Documentación de Spring

En la figura 6 podemos ver los diferentes módulos que nos ofrece spring framework que se pueden integrar a la aplicación web.

Aop (Programación Orientada a Aspectos)

Se trata de un módulo que nos permitirá utilizar el paradigma de Programación Orientada a Aspectos (Aspect Oriented Programming). Este paradigma permite mejorar la modularización y separar las responsabilidades. De esta forma, podemos separar las

funcionalidades comunes, que se utilizan transversalmente a lo largo de toda la aplicación, de aquellas que son propias de cada módulo.

Data Access/Integración

Es un módulo muy importante y está formado por múltiples submódulos, facilita la programación simplificando el acceso y persistencia de los datos. Spring Data nos proporciona soporte para usar base de datos relacionales (JDBC), ORMs (como por ejemplo JPA, Hibernate, etc.) e incluso modelos de persistencia de base de datos no relacionales (NoSQL).

Web

Este módulo nos permitirá implementar el patrón Modelo-Vista-Controlador (MVC) de una manera sencilla y limpia, haciendo uso de forma transparente también de otros patrones de diseño, como FrontController. De esta forma, podemos separar limpiamente la lógica de negocio de la presentación de los datos y el acceso a los mismos. Además de aplicaciones que implican el uso de vistas y formularios, también podremos crear servicios web (por ejemplo, al estilo REST) de una forma sencilla y rápida, además de integrarse fácilmente a todas las demás características de spring framework.

2.3.2.2. Inversión de control

Cuando hablamos de Spring Framework, hablamos de Inversión de Control (IoC), tiene que ver con la dirección que adopta el software, de esta manera se realiza un cambio radical a lo que se venía realizando en la programación secuencial, es decir; cuando se programa se definen las secuencias de operaciones y el fuerte acoplamiento que mantienen

las clases para llegar al resultado final, sin embargo, con el framework de spring se realiza lo contrario, es el framework quien toma el control de las operaciones desde la creación hasta la eliminación del objeto.

El componente Spring Framework Inversión of Control (IoC) aborda esta preocupación al proporcionar un medio formalizado de componer componentes dispares en una aplicación completamente funcional lista para usar. Spring Framework codifica patrones de diseño formalizados como objetos de primera clase que puede integrar en sus propias aplicaciones. Numerosas organizaciones e instituciones utilizan Spring Framework de esta manera para diseñar aplicaciones robustas y fáciles de mantener (Spring, 2016).

Podemos decir que la Inversión de Control se implementa con la necesidad de redefinir la manera de cómo se venía codificando, brindándonos componentes estandarizados para seguir un modelo de desarrollo ágil y flexible.

A continuación, se muestra un ejemplo referencial a fin de entender cómo se realiza un programa que no tiene la inversión de control en su estructura de desarrollo, de esta manera se sigue la secuencia del código establecido por el programador y el fuerte acoplamiento de las clases para llegar el resultado final.

Figura7

Acoplamiento de Clase - Ejemplo

```

package demo_programasecuencial;
/**
 *
 * @author Edgar
 */
public class VotoPresencial {
    public void mostrar(){
        System.out.println
        ("Tú voto se efectuó de manera Presencial");
    }
    public VotoPresencial() {
    }
}

package demo_programasecuencial;
/**
 *
 * @author Edgar
 */
public class Persona {
    VotoPresencial voto = new VotoPresencial();
    public void muestraTipoVotacion(){
        voto.mostrar();
    }
}

2 package demo_programasecuencial;
3 /**
4  *
5  * @author Edgar
6  */
7 public class Demo_programaSecuencial {
8     public static void main(String[] args) {
9
10         Persona persona=new Persona();
11         persona.muestraTipoVotacion();
12     }
13 }
14
15 }
16

Output - demo_programaSecuencial (run) x
run:
Tú voto se efectuó de manera Presencial
BUILD SUCCESSFUL (total time: 0 seconds)

```

Fuente: Elaboración propia

La figura 7 muestra un programa que representa el tipo de votación que ha realizado una persona, para ello se implementa la clase VotoPresencial, Persona y el main para la ejecución, además se puede ver la secuencia que se tiene que seguir y el fuerte acoplamiento de las clases mediante el operador **new**. Podemos decir que el programa está cumpliendo con su objetivo de conocer de qué manera la persona ha emitido su votación, pero que pasaría si la modalidad de votación cambiará a voto electrónico, o voto por correo o alguna

otra modalidad de votación, sin duda sería un problema de implementación debido a la falta de flexibilidad del programa.

Para logra un desacoplamiento de nuestras clases y hacerlo de manera más flexible usaremos el patrón de diseño de inyección de dependencias (IoC) y agregaremos una interfaz, allí se implantarán los métodos comunes entre los tipos de votos que puedan existir.

2.3.2.3. Inyección de dependencia

La inyección de dependencia (DI) es un proceso mediante el cual los objetos definen sus dependencias, es decir, los otros objetos con los que trabajan, solo a través de argumentos de constructor, argumentos a un método de fábrica o propiedades que se establecen en la instancia del objeto después de que se construye o devuelve de un método de fábrica. Luego, el contenedor inyecta esas dependencias cuando crea el bean. Este proceso es fundamentalmente el inverso, de ahí el nombre Inversión de control (IoC), del propio bean que controla la instanciación o ubicación de sus dependencias por sí solo mediante la construcción directa de clases, o el patrón de localizador de servicios. (Spring, 2016)

Podemos decir que la inyección de dependencias (DI) permite establecer un vínculo con otro objeto que contiene todas las operaciones para obtener sus acciones y tiene una relación con la inversión de control (IoC) para el tratado de los bean.

Figura8

Inyección de Dependencia - ejemplo

```

package demo_programasecuencial;
/**
 *
 * @author Edgar
 */
public class VotoPresencial implements IVoto {
    public VotoPresencial() {
    }
    @Override
    public void mostrar() {
        System.out.println
            ("Tú voto se efectuó de manera Presencial");
    }
}

package demo_programasecuencial;
/**
 *
 * @author Edgar
 */
public class VotoVirtual implements IVoto {
    public VotoVirtual() {
    }
    @Override
    public void mostrar() {
        System.out.println
            ("Tú voto se efectuó de manera Virtual");
    }
}

package demo_programasecuencial;
/**
 *
 * @author Edgar
 */
public class Persona {
    private final IVoto voto;
    public Persona(IVoto voto) {
        this.voto=voto;
    }
    public void muestraTipoVotacion() {
        voto.mostrar();
    }
}

package demo_programasecuencial;
/**
 *
 * @author Edgar
 */
public class Demo_programaSecuencial {
    public static void main(String[] args) {
        Persona persona=new Persona(new VotoPresencial());
        persona.muestraTipoVotacion();
        Persona per=new Persona(new VotoVirtual());
        per.muestraTipoVotacion();
    }
}

```

1
2
3
4

```

tput - demo_programaSecuencial (run) x
run:
Tú voto se efectuó de manera Presencial
Tú voto se efectuó de manera Virtual
BUILD SUCCESSFUL (total time: 0 seconds)

```

Fuente: Elaboración propia

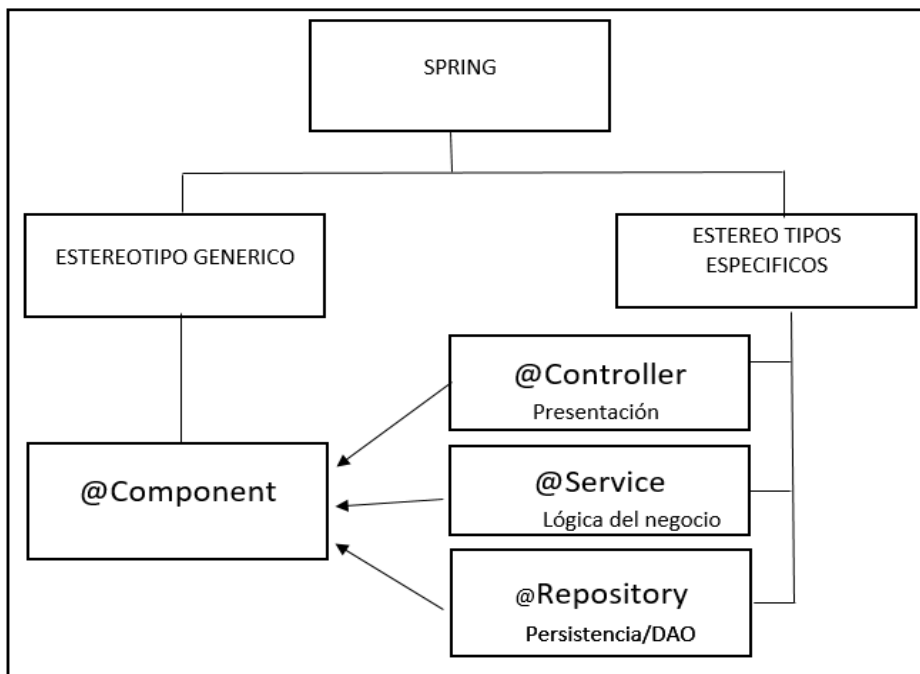
En la figura 8 podemos ver que estamos implementado la interface IVoto a las clases VotoVirtual y VotoPresencial, de esta manera estamos utilizando su método genérico mostrar () para replicarlo en cada una de las clases mencionadas, así mismo ahora vemos que la clase Persona se relaciona con la interface IVoto y ya no con las clases VotoVirtual y VotoPresencial, rompiendo el fuerte acoplamiento que tenían en la figura xx anterior.

2.3.2.4. Estereotipos

A partir de Spring 2.5 existen anotaciones de estereotipos que simplifican la configuración de clases de nuestro proyecto, podemos decir que un estereotipo en spring es una clase o interface que recibe una inyección mediante anotaciones con la finalidad de que spring trate estas clases como un beans o componente configurado para su fácil detección y administración, estas anotaciones son:

Figura9

Estereotipos de Spring



Fuente: Elaboración propia

@Component. _ Es una anotación genérica que puede ser implementado en cualquier clase para que spring pueda considerarla como un beans, la documentación de spring recomienda utilizar la anotación correspondiente para cada clase específica y evitar el uso de anotaciones genéricas.

@Repository. _ Es una anotación generalmente implementada en el patrón DAO (Data Access Object) el cual permite separar la lógica del acceso a datos.

@Service. _ Está anotación se encarga de mantener la lógica del negocio y gestionar las diversas fuentes de la información.

@Controller. _ Se encarga de controlar y gestionar las peticiones que vienen del frontend, controla entradas y salidas de información, se emplea para la capa de presentación.

Spring ha incrementado sus anotaciones para utilizar en cada etapa del proyecto, la anotación @Service se utiliza para los servicios, @Repository para la persistencia y @Controller para la presentación, estas anotaciones son estereotipos específicos que aportan una mayor adaptabilidad en comparación de la anotación @Component que tiene un comportamiento más genérico (Sping, 2020).

2.3.2.5. Anotaciones

Spring proporciona diferentes tipos de anotaciones para diferentes capas de la aplicación, a continuación, mencionaremos las utilizadas en el presente proyecto.

Anotaciones de persistencia

@Entity: Esta anotación se aplica a la clase para indicarle a hibernate que se comporte como una entidad que se tiene que persistir.

@Table: Esta anotación trabaja con la anotación @Entity, pues una vez que ya tenemos la entidad, tendremos que darle un nombre al objeto, mediante un parámetro que mapea el nombre original de la tabla de la base de datos.

@Id: Se utiliza para indicar que la propiedad que tiene esta anotación es la llave primaria.

@GeneratedValue: Es anotada el campo primario para generar su valor de la entidad para que sea auto incrementable.

@Column: Se utiliza para nombrar o renombrar al nombre que tendrá cada propiedad del objeto. Si no se especifica esta anotación, se tomará por defecto el nombre de la variable.

@JoinColumn: Se utiliza para hacer referencia a la tabla vinculada, actúa como llave foránea.

@NotEmpty: Utilizados para propiedades de cadena, impide propiedades vacías.

@OneToMany: Establece la relación entre dos entidades según su cardinalidad con otra entidad, en este caso la relación establecida es de uno a muchos.

@ManyToOne: Esta anotación se declara cuando la relación es de muchos a uno entre dos entidades, básicamente sería el retorno de @OneToMany.

@OneToOne: Esta anotación establece la relación de uno a uno entre dos entidades.

@Query: Esta anotación permite utilizar consultas JPQL y consultas de SQL nativas.

@Temporal: Usamos esta anotación para persistir y dar formato a nuestro campo de fecha.

Anotaciones de seguridad

@EnabledGlobalMethodSecurity: Esta anotación se utiliza para poder habilitar la seguridad a los métodos de nuestro controlador mediante roles.

@EnabledWebSecurity: Esta anotación se utiliza para agregar soporte a Spring Boot para poder utilizar Spring Security.

@Secured: Esta anotación la utilizamos para establecer el nombre de rol que tendrá cada método de nuestro controlador.

Anotaciones de presentación

@RequestMapping: Se utiliza en la capa controladora para para peticiones GET de HTTP.

@GetMapping: Esta anotación dentro del proyecto tiene la misma funcionalidad que la anotación

@RequetMapping: Tiene una funcionalidad similar a la anotación @GetMapping.

@PostMapping: Esta anotación se utiliza para peticiones POST de HTTP en nuestra capa controladora.

@ResponseBody: Esta anotación indica que un parámetro de método debe de retornar en el cuerpo de respuesta vinculado a la solicitud HTTP, no se necesita una vista formal para mostrar la información.

Otras anotaciones

@Autowired: Es probablemente una de las anotaciones más utilizada en Spring, permite hacer la inyección de dependencias con otras entidades.

@Configuration: Se declara esta anotación cuando tenemos una clase que contiene métodos que serán tratados como beans.

2.3.3. Aplicación web y herramientas

Los sistemas web hoy en día son muy requeridos y utilizados por diferentes empresas a nivel mundial, y la razón obedece a su gran capacidad para interactuar desde cualquier parte del mundo o el bajo costo en la utilización de hardware para entrar en producción, un sistema web es una plataforma que nos permite conectarnos desde cualquier dispositivo para realizar diferentes actividades sin necesidad de descargar la aplicación, existen diferentes herramientas informáticas que nos permiten generar una aplicación web, es este proyecto utilizaremos Spring Tool Suit (STS).

2.3.3.1. Spring Tool Suite -versión 4.9.0 (Sts)

Spring Tool Suite es un ID de desarrollo basado en Java EE de Eclipse, creado por la comunidad de SpringSource basado en plantillas de Spring. Cuenta con plugins incluidos que facilitan la creación del proyecto y viene con Maven integrado para la descarga automática de las dependencias o librerías que necesita el proyecto.

Diseñado para desarrollar aplicaciones empresariales utilizando Spring Framework y Spring Boot, la nueva generación de Spring Tools brinda soporte de desarrollo de clase mundial para sus aplicaciones Spring. Nuestras herramientas tienen un profundo conocimiento de Spring integrado (Spring, s.f.)

2.3.3.1.1. Requisitos De Instalación Sts

Para poder instalar y utilizar el ID Spring Tool Suit, primero tenemos que tener instalado y configurado el JDK 8 o superior de java luego tenemos que ir a su página oficial <https://spring.io/tools>, elegir el sistema operativo adecuado, descargar e instalar.

Figura10

Sistema operativos para Spring Tool Suit



Una vez descargado el archivo según nuestro sistema operativo, procederemos a correr el ejecutable y nos muestra una pantalla similar a la que mostramos a continuación, donde podremos crear un nuevo proyecto.

Figura11

Generar Nuevo Proyecto Spring Tool – Paso 1

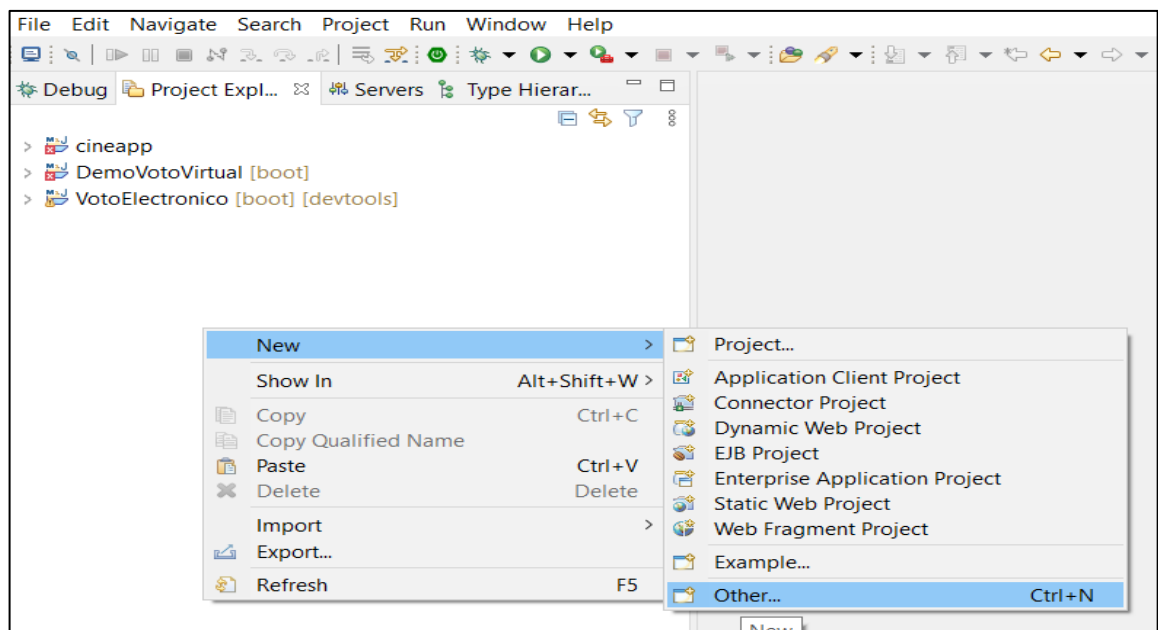


Figura12

Generar Nuevo Proyecto Spring Tool – Paso 2

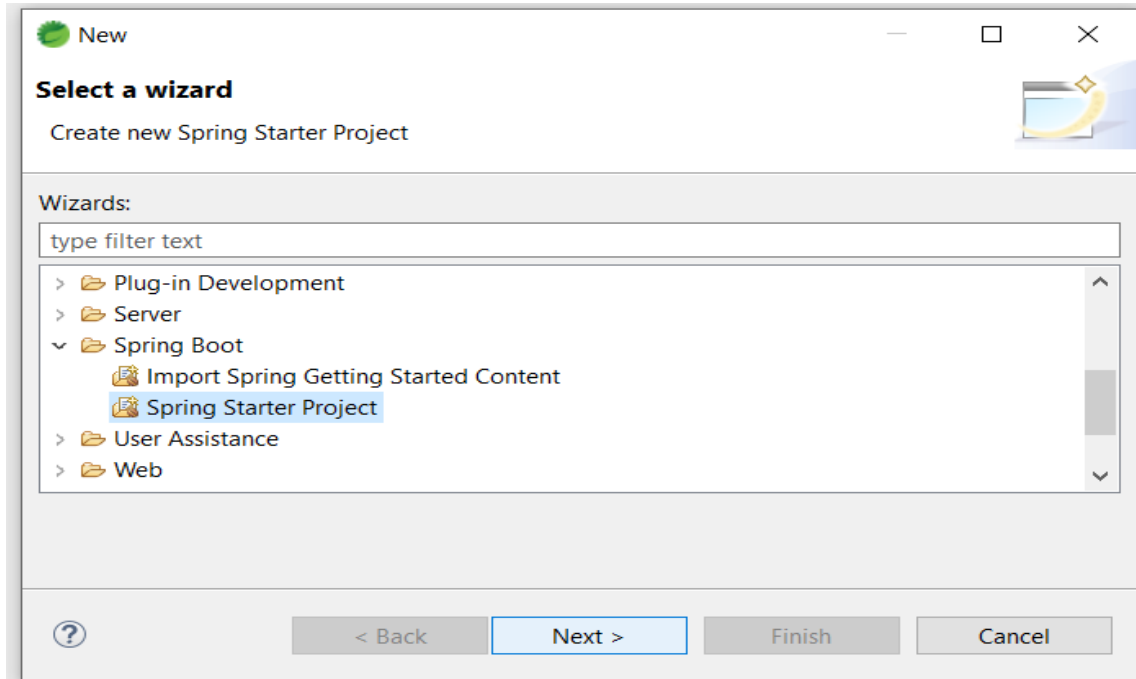
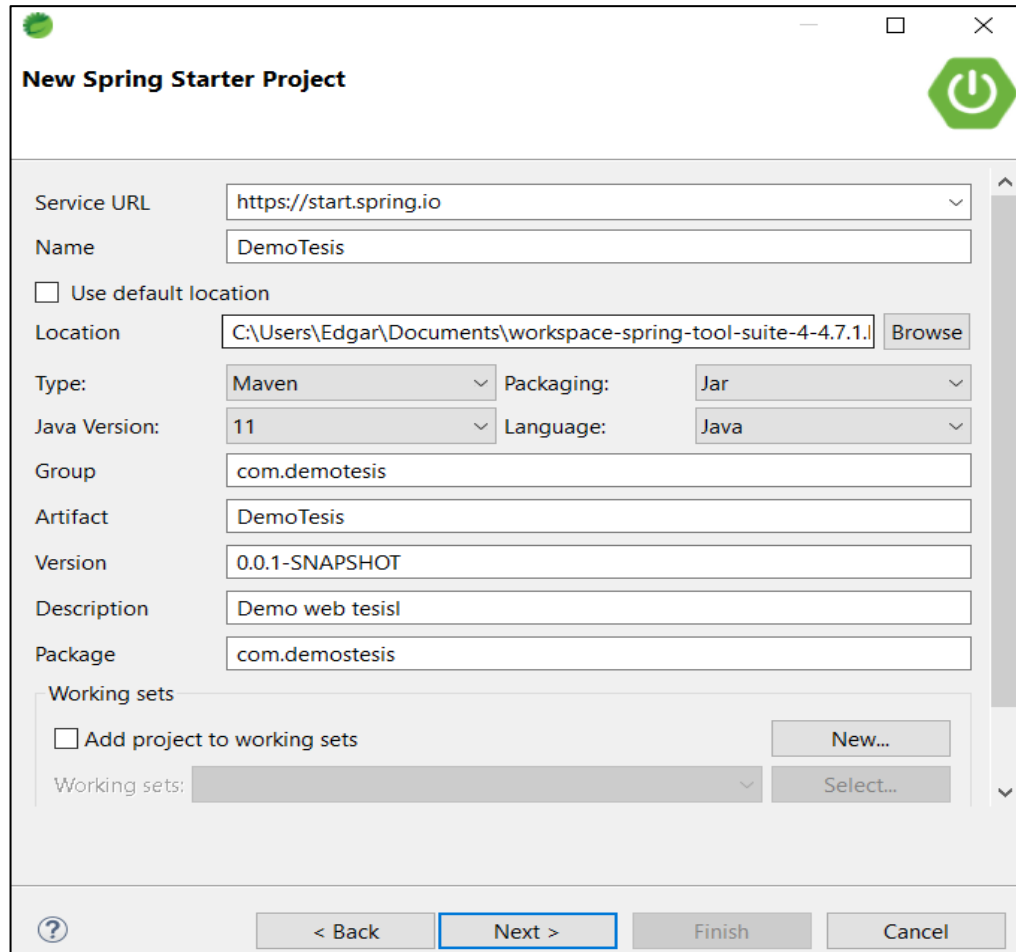


Figura13

Generar Nuevo Proyecto Spring Tool – Paso 3



The screenshot shows the 'New Spring Starter Project' dialog box. The fields are filled with the following information:

- Service URL: <https://start.spring.io>
- Name: DemoTesis
- Location: C:\Users\Edgar\Documents\workspace-spring-tool-suite-4-4.7.1. (with a 'Browse' button)
- Type: Maven
- Packaging: Jar
- Java Version: 11
- Language: Java
- Group: com.demotesis
- Artifact: DemoTesis
- Version: 0.0.1-SNAPSHOT
- Description: Demo web tesis
- Package: com.demostesis

At the bottom, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'. There is also a 'Working sets' section with an 'Add project to working sets' checkbox and 'New...' and 'Select...' buttons.

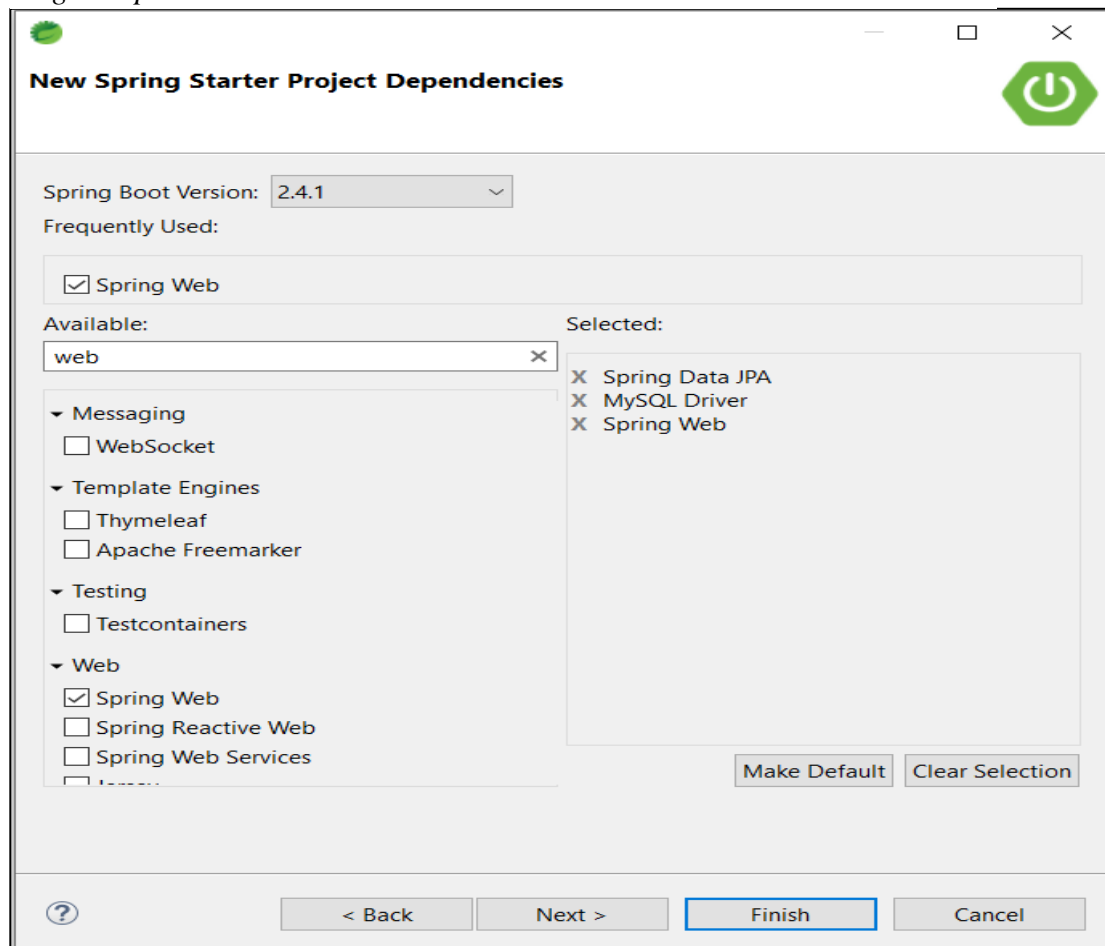
En la figura 13 nos muestra una interface con la siguiente información:

- Service URL. _ Para utilizar STS hemos tenido que descargar el de manera local, sin embargo, también podemos crear un proyecto sin tener que descargar el STS, para ello tenemos que ir a la ruta que indica este campo <https://start.spring.io> y nos permitirá crear un esqueleto de nuestro proyecto para luego poder ser importado a nuestro STS local, actúa como alternativa de crear una estructura de proyecto de manera rápida y eficaz..
- Name. _ Nombre que va tener el proyecto.
- Location. _ Ruta donde se guardará el proyecto.

- Type. _ Permite elegir la herramienta que se encargará de hacer la configuración e importaciones de dependencias y pueden ser de 2 tipos, maven y gradle.
- Group. _ En este campo de debe de digitar el nombre o identificador de tu empresa, este nombre formará parte de los paquetes que se implemente en el proyecto.
- Packaging. _ Este campo es para elegir el formato del proyecto, este puede ser un formato Jar o formato War.
- Lenguaje. _ Aquí podemos elegir el lenguaje de programación que usaremos, de entre las alternativas que tenemos se encuentran; Java, Kotlin y Groovy.

Figura 14

Elegir Dependencias Paso 4



En la figura 14 podemos ver un interfaz que nos permitirá elegir las dependencias que tendrá nuestro proyecto, en este caso estamos implementando un proyecto WEB con persistencia para JPA y base de datos MYSQL. Una vez seleccionados los componentes que tendrán nuestro proyecto, daremos clic en finalizar y todas las dependencias se agregarán de manera automática a nuestro proyecto dejándonos listo para empezar a programa.

2.3.3.2. Gestores de bases de datos

Un sistema gestor de base de datos o SGBD (aunque se suele utilizar más a menudo las siglas DBMS procedentes del inglés, ***Data Base Management System***) es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos. (Nevado Cabello, 2010).

El gestor de base de datos tiene como función principal guardar toda la información para ser mostrada mediante una interfaz, existen diferentes gestores de bases de datos y se clasifican de la siguiente manera:

2.3.3.2.1. Gestor de bases de datos relacional

En 1970 se propusieron por primera vez las bases de datos relacionales y las teorías subyacentes [3], entre las que se destaca el modelo de base de datos relacional, que implicó un cambio radical en el manejo de la información apoyándose en operaciones de conjuntos que combinan tablas de datos separadas (o relaciones) para producir un conjunto de respuestas. Las consultas se especifican utilizando el lenguaje de consulta estructurado SQL (por las siglas en inglés de Structured Query Language), soportado en el álgebra relacional, y que permite a un usuario expresar su consulta en forma declarativa, sin ningún tipo de instrucciones detalladas de programación. (Castro et al. 2012).

Un gestor de base de datos relacional básicamente en la unión de dos o más entidades (tablas de base de datos), donde se tiene que identificar cual de las entidades será la entidad principal para así tener una referencia de ella en la entidad secundaria, estableciendo de esta manera una relación que se debe de cumplir para el grabado de la información.

La normalización en bases de datos relaciones es un procedimiento estándar que se tiene que cumplir para evitar la redundancia de información, para ello se tienen que cumplir diferentes formad de normalización de los datos.

Para Microsoft (2021), las normalizaciones de los datos se dan de la siguiente manera y nos muestra un ejemplo de cada uno de las normas:

a. Primera forma normal:

- Debe existir clave única en cada tabla
- No debe de existir redundancia en cada tabla

b. Segunda forma normal:

- Crear tablas independientes para conjuntos de valores que se aplican a varios registros.
- Si una tabla que puede ser relacionada con una o más tablas, esta debe de una tabla independiente

c. Tercera forma normal:

- Se debe de depurar aquellos campos que no tengan dependencia de la clave

2.3.3.2.2. Gestores de bases de datos más utilizados

A continuación, se muestran algunos de los gestores de bases de datos más utilizados.

- Mysql
- Microsoft - SQL
- PostgreSQL
- Oracle

Existen otros gestores de base de datos relacionales, solo se ha mencionado aquellos gestores que para el autor de la presente tesis son los más importante. He de mencionar que el proyecto de voto electrónico utilizará el gestor de base de datos Mysql por su fácil acoplamiento al sistema operativo Linux. Sin embargo, el presente proyecto utilizará mapeo relacional de objetos (ORM), lo que permitirá al sistema de voto electrónico acoplarse a cualquier gestor de bases de datos.

2.3.3.3. Patrón de arquitectura

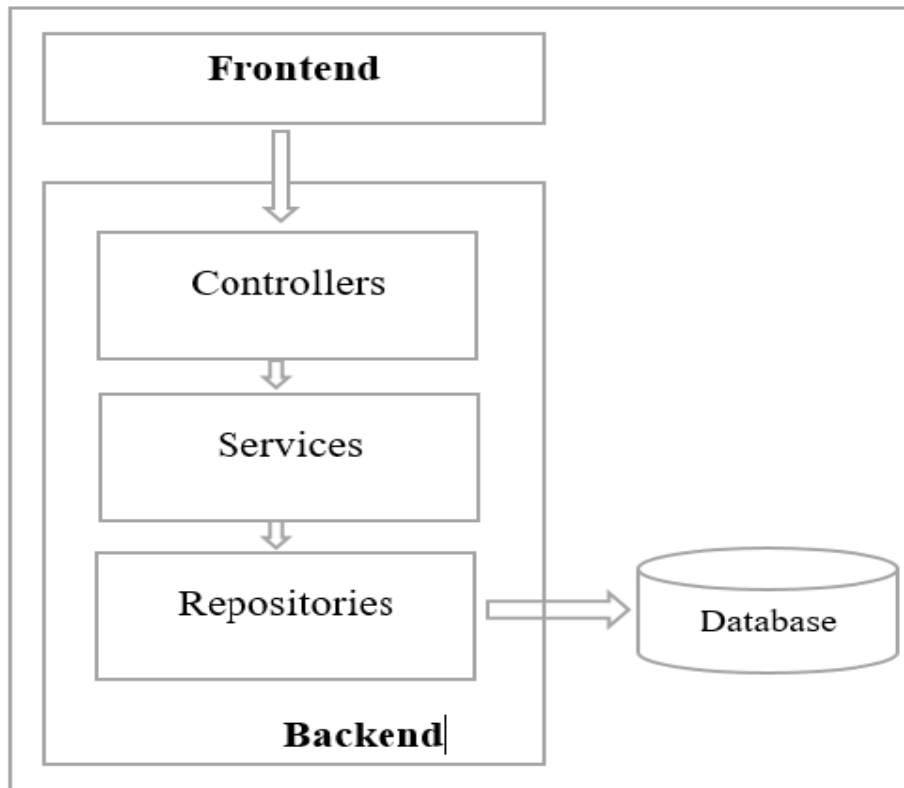
Modelo vista controlador (MVC) tradicional

El patrón de diseño Modelo-Vista-Controlador se utiliza para el diseño de aplicaciones con interfaces complejas. La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad de las partes (Fernández et al., 2012).

El patrón MVC es un modelo de arquitectura de software a seguir que permite interactuar sus tres componentes (Modelo, la vista y el controlador) sin que estos componentes se encuentren juntos, separando de esta manera la funcionalidad y dependencias de cada uno de ellos.

Figura 15

Patrón Modelo Vista Controlador MVC



Fuente: Elaboración propia

En la figura 15 podemos ver como spring utiliza sus anotaciones de estereotipos para generar la capa de backend basado en el patrón de arquitectura MVC (modelo vista controlador), este patrón de arquitectura puede utilizarse para aplicaciones rest o aplicaciones web y su construcción se simplifica utilizando spring boot.

Frontend

Representa la capa de presentación, es la que se muestra al usuario para su interacción, esta puede ser mediante html, jsp, Thymeleaf, json, etc.

El frontend es la interface que se le mostrara al usuario quien será el que interactúe con aplicación, es fundamental conocer las funciones que tiene que realizar el usuario con la aplicación para poder

brindarle una interacción que satisfaga su necesidad en las diversas tareas que tiene que realizar de lado cliente de la aplicación, conociendo estos requerimientos sabremos qué modelo de frontend se tendrá que implementar (Marini, 2012).

Backend

Controller: Controla entradas y salidas de información, se encarga de recibir las peticiones del frontend, representa la capa de presentación de la aplicación mediante una fallada amigable para el usuario.

Services: Se encarga de la lógica de negocio, se encarga de gestionar la fuente de información permitiendo la comunicación con uno o más repositorios de diferentes fuentes de datos.

Repositories:

Esta capa es referida al patrón de diseño DAO (Data Access Object) Se encarga de generar acciones a la base de datos, es la capa encargada establecer la conexión con la base de datos para realizar el crud (create, read, update y delete) del sistema web.

2.3.3.4. Spring boot

Spring Boot le ayuda a crear aplicaciones independientes basadas en Spring de grado de producción que puede ejecutar. Tomamos una visión obstinada de la plataforma Spring y las bibliotecas de terceros, para que pueda comenzar con el mínimo esfuerzo. La mayoría de las aplicaciones Spring Boot necesitan muy poca configuración de Spring (Spring, 2020).

Spring boot es un producto de la familia de spring, que tiene por objetivo simplificar la creación de aplicaciones y servicios, cuenta con plugins ya integrados que se encargan de la

configuración de librerías o dependencias que requiere el proyecto, además de traer ya configurado un servidor web.

2.3.3.4.1. Maven

Maven, una palabra yiddish que significa acumulador de conocimiento, comenzó como un intento de simplificar los procesos de construcción en el proyecto Jakarta Turbine. Había varios proyectos, cada uno con sus propios archivos de compilación Ant, que eran todos ligeramente diferentes. Los JAR se registraron en CVS. Queríamos una forma estándar de construir los proyectos, una definición clara de en qué consistía el proyecto, una forma fácil de publicar información del proyecto y una forma de compartir los archivos JAR en varios proyectos (Maven, 2021).

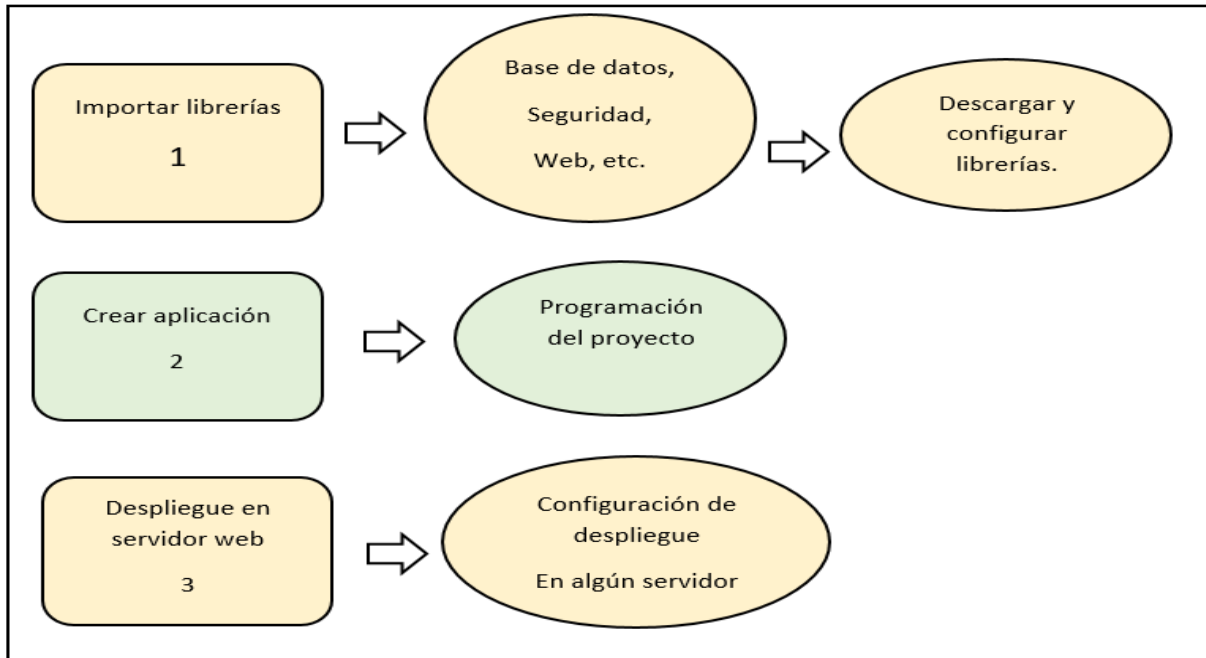
Básicamente Maven es una herramienta de código abierto para gestionar proyectos de software basado en el concepto POM (Project Object Model), se encarga de centralizar, gestionar y configurar automáticamente todas las dependencias que necesita el proyecto, facilitando enormemente el proceso de configuración manual de las dependencias o librerías y sus versiones que requería el proyecto, Maven es el aliado ideal de Spring boot para la construcción de un software.

Esquema para crear un proyecto sin Maven

En términos generales se siguen tres secuencias para la creación de una aplicación web según la imagen referencial que se muestra a continuación.

Figura16

Esquema usando Maven



Fuente: Elaboración propia

La figura 16 muestra tres secuencias que se deben de considerar para crear una aplicación web, en el paso 1 se tienen que descargar todas aquellas librerías o dependencias para nuestro proyecto, por ejemplo si queremos utilizar base de datos, debemos de descargar y configurar el Jar adecuado a la versión y lo mismo si queremos utilizar seguridad u otra funcionalidad que requiera alguna dependencia, en el paso 2 es para crear el proyecto y la dedicación de solo programar los módulos que hagan funcionar el mismo y en el paso 3 tendríamos que configurar algún servidor web que permita realizar el despliegue de nuestro proyecto.

Pom.xml

Es un fichero XML que actúa como contenedor principal de todas las referencias, dependencias, fuentes, versiones, etc., que ha configurado Maven para que el proyecto pueda ejecutarse satisfactoriamente.

Figura17

Pom Xml que contiene las dependencias del proyecto

```
*VotoElectronico/pom.xml ☒
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6   <parent>
7     <groupId>org.springframework.boot</groupId>
8     <artifactId>spring-boot-starter-parent</artifactId>
9     <version>2.2.6.RELEASE</version>
10    <relativePath /> <!-- lookup parent from repository -->
11  </parent>
12  <groupId>com.implementalo</groupId>
13  <artifactId>VotoElectronico</artifactId>
14  <version>0.0.1-SNAPSHOT</version>
15  <packaging>war</packaging>
16  <name>VotoElectronico</name>
17  <description>Voto electronico para asambleas</description>
18  <properties>
19    <java.version>1.8</java.version>
20  </properties>
21  <dependencies>
22    <dependency>
23      <groupId>mysql</groupId>
24      <artifactId>mysql-connector-java</artifactId>
25      <scope>runtime</scope>
26    </dependency>
27    <dependency>
28      <groupId>org.springframework.boot</groupId>
29      <artifactId>spring-boot-starter-data-jpa</artifactId>
30    </dependency>
31    <dependency>
32      <groupId>org.springframework.boot</groupId>
33      <artifactId>spring-boot-starter-thymeleaf</artifactId>
34    </dependency>
35    <dependency>
36      <groupId>org.springframework.boot</groupId>
37      <artifactId>spring-boot-starter-web</artifactId>
38    </dependency>
```

Fuente: Elaboración propia

La figura 17 muestra el archivo de configuración pom.xml, el cual envuelve en sus etiquetas las referencias del proyecto. Existen 3 puntos clave en este archivo de configuración:

<parent>: Hace referencia que es un proyecto de Spring Boot e indica la versión que se está utilizando, si quisiera cambiar de versión solo deberá indicar o modificar la versión actual y los archivos y dependencias de actualizaran.

<dependencies>: Hace referencia a las dependencias que se están utilizando en el proyecto, si quisiera agregar alguna otra herramienta a su proyecto que requiera a su proyecto, tendrá que agregar aquí sus dependencias.

<build>: Es el complemento que proporciona las librerías necesarias que ayudan a que su proyecto se pueda ejecutar directamente sin implementarlo en un servidor web. Ayuda a crear un archivo jar que puede ser ejecutable.

Es importante mencionar que para que la descarga de complementos y dependencias se realicen de manera satisfactoria, se tiene que tener conexión a internet para que Maven puede extraer los archivos y dependencias.

2.3.3.5. Persistencia de datos

2.3.3.5.1. Aplicaciones de persistencia Java (JPA)

JPA es una especificación de java para administrar el mapeo relación de objeto, brinda un estándar de especificaciones para convertir entidades de base de datos en objetos y persistirlo para luego abstraerse de la base de datos. Brinda el estándar de especificación para poder relacionar los objetos según su cardinalidad, actualmente podemos utilizar anotaciones para simplificar estas relaciones.

Aplicando las especificaciones de JPA, brinda una capa de seguridad ante los ataques de SQL injection ya que las consultas se realizan sobre los objetos y no sobre la base de datos, rapidez para las consultas y flexibilidad de conexión con cualquier base de datos relacional. Cuenta con su lenguaje de consultas de persistencia de java (JPQL) para hacer las consultas sobre los objetos de la aplicación.

2.3.3.5.2. *Hibernate (ORM)*

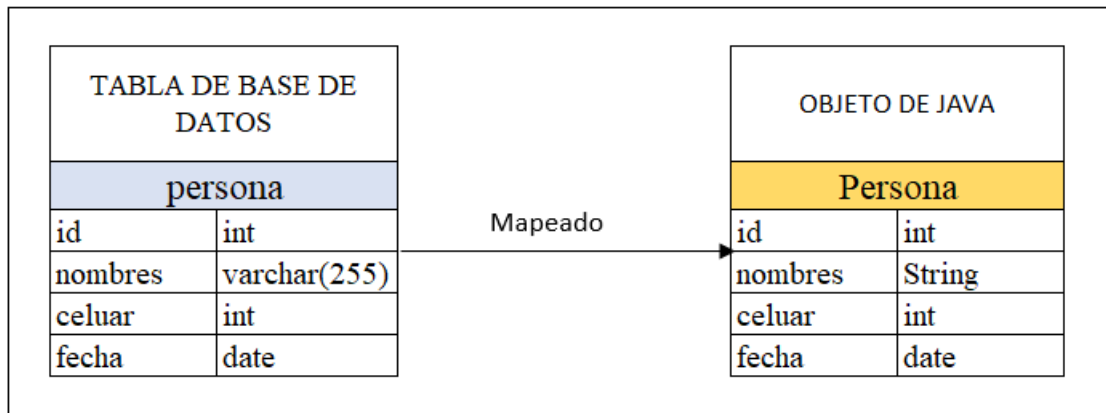
Para que JPA pueda funcionar necesitamos de una herramienta que se encargue de implementar todas las especificaciones encargadas de la persistencia y la relación de los objetos, en este proyecto usaremos Hibernate por ser uno de los ORM más utilizados actualmente.

Hibernate se encarga del mapeo de las clases de Java a las tablas de la base de datos y de los tipos de datos de Java a los tipos de datos de SQL. Además, proporciona funciones de consulta y recuperación de datos. Puede reducir significativamente el tiempo de desarrollo que de otro modo se gastaría con el manejo manual de datos en SQL y JDBC. El objetivo de diseño de Hibernate es aliviar al desarrollador del 95% de las tareas comunes de programación relacionadas con la persistencia de datos al eliminar la necesidad de procesamiento de datos manual y hecho a mano mediante SQL y JDBC. Sin embargo, a diferencia de muchas otras soluciones de persistencia, Hibernate no le oculta el poder de SQL y le garantiza que su inversión en tecnología relacional y conocimiento es tan válida como siempre (Hibernate, 2021).

Hibernate es una herramienta de mapeo relaciona que nos permite mapear la estructura de una base de datos relacional con finalidad de convertir las tablas en objetos y poderlos manipular desde nuestra aplicación y dejar de lado las dependencias de las tablas de las bases de datos. Supongamos que tenemos una base datos en SQL y de pronto el administrador nos dice que se tiene que migrar a Mysql porque no quiere renovar la licencia, ese cambio implicaría reestructurar y configurar toda la base de datos por medio de herramientas externas para luego revisar el funcionamiento de los procedimientos almacenados y funciones, hacer este cambio implica gastos de recursos.

Figura18

Mapeo entre Tabla de base de datos y Objeto Java



Fuente: Elaboración propia

La figura 18 muestra el esquema de cómo se puede mapear una tabla de una base de datos con hibernate, podemos que ver que cada campo es replicado tomando en cuenta el tipo de dato para luego hacer todas nuestras consultas sobre el objeto. Para convertir esta tabla en objeto tenemos que hacer uso de las anotaciones de persistencia indicada en la sección de anotaciones.

2.3.3.5.3. Relaciones de objetos

Para poder relacionar los objetos, hibernate se basa en la especificación de JPA. Este procedimiento es básicamente copiar el diseño de nuestro diagrama de entidad relación y aplicarlo a los objetos e ir relacionando, para ello se tiene que aplicar los diferentes tipos de asociaciones que se mencionan a continuación.

- **One to One (Unidireccional):** Un objeto tiene una referencia de otro objeto.
- **One to One (Bidireccional):** Los dos objetos tienen referencia unos de otro.
- **One to Many:** Un objeto tenga una lista de otros objetos

- **Many to Many:** Un objeto A tenga una lista de otros objetos B y también que el objeto B a su vez tenga la lista de objetos A

Una de las formas de acceder a estos relacionados con hibernate es utilizar su lenguaje de consulta de hibernate (HQL) sobre los objetos persistido, este lenguaje de consulta permite generar select con uno o más objetos mediante joins (Hibernate, s.f.).

2.3.3.5.1. Spring data Jpa

Implementar una capa de acceso a datos de una aplicación ha sido engorroso durante bastante tiempo. Se debe escribir demasiado código repetitivo para ejecutar consultas simples, así como para realizar la paginación y la auditoría. Spring Data JPA tiene como objetivo mejorar significativamente la implementación de capas de acceso a datos al reducir el esfuerzo a la cantidad que realmente se necesita. Como desarrollador, escribe las interfaces de su repositorio, incluidos los métodos de búsqueda personalizados, y Spring proporcionará la implementación automáticamente (Spring, s.f.).

Spring Data Jpa es parte de los módulos Spring Data quien a su vez forma parte de Spring framework, Spring Data JPA simplifica la creación de métodos repetidos en la capa de acceso a datos (DAO), extendiendo de interfaces que traen repositorios con interfaces que contienen métodos ya creados para que sean reutilizado por cualquier objeto, acortando el camino de la programación.

A modo de ejemplo, supongamos que se va implementar una nueva clase u objeto de nombre Persona a su proyecto, usted crea sus propiedades y luego tiene que crear sus métodos CRUD de ese objeto en la capa DAO, empiezas a crear el update (), delete (), insert () y algún select. Luego

esos métodos son invocados por el controlador para ser llevado a la vista, ahora imagine que también tiene que agregar la clase Vehículo, tiene volver a crear nuevamente su CRUD para la clase Vehículo y así sucesivamente con las otras clases que tenga que implementar. El ejemplo mencionado sería la forma tradicional como se venían realizando el CRUD en la capa Dao, Utilizando Spring Data Jpa facilita realizar estas acciones ya que genera de manera automática el CRUD para ser utilizado en cualquier clase.

Además de generarte el CRUD de acciones de manera automática, te ofrece palabras reservadas que actúan como un where de SQL para consultar un objeto de manera simple o consultar objetos persistidos relacionados mediante su lenguaje de consultas de persistencia de java (JPQL).

A continuación, se muestra algunas de las palabras reservadas que se pueden utilizar en el JPQL.

- **Distinct**
- **Or**
- **And**
- **Between**
- **Like**
- **Null**
- **IsnNull**

Existen más palabras reservadas que se pueden utilizar y que pueden ser revisadas en la documentación de Spring Data Jpa.

Figura19

Consultas con Spring Data Jpa

Persona **findBy**Dni (String dni)
Persona **findBy**Email**And**Nombre (String email, Spring nombre)

Fuente: Elaboración propia

En la figura 19 se puede apreciar la manera referencial de como se pueden emplear las palabras reservadas para hacer una búsqueda sobre un objeto, en el primer metodo se realiza una búsqueda por el dni de la persona y el el segundo metodo se realiza una búsqueda por email y nombre del la persona.

Figura 20

Jpa Repository - Crud Repository

```
@Repository
public interface SocioDao extends CrudRepository<Socio, Long> {
    @Query("select v from Voto v where v.socio.id=?1 and v.propuesta.id=?2 and estado=1" )
    List<Voto>findBySocioAndPropuesta(Long socioId, Long PropuestaId);

    Socio findByEmail(String email); // G
}
```

Fuente: Elaboración propia

En la figura 20 vemos que la interface SocioDao, puede ser extendida por la interfaz CrudRepository o JpaRepository, esta interfaz contiene el CRUD y otros métodos ya creado. Además, se puede apreciar que existe la anotación @Query en el método

findBySocioAndPropuesta, esta interfaz permite generar su consulta para relacionar entidades, en este caso se está relacionado la entidad Voto con la entidad socio y se le esta enviado dos parámetros para que nos devuelva una lista. También existe un método findByEmail con un parámetro para devolver al socio que le corresponda ese email. Es importante mencionar que JPQL también soporta consultas nativas de SQL.

Figura 21

Inyección de dependencia con Spring

```
11 @Service
12 public class SocioServiceImpl implements SocioService {
13
14     @Autowired
15     private SocioDao _socioDao;
16
17
18     @Override
19     public Socio buscarPorDni(String dni) {
20         // TODO Auto-generated method stub
21         return _socioDao.;
22     }
23
24
25     @Override
26     public void guardarSocio(Socio socio) {
27         // TODO Auto-generated method stub
28         _socioDao.save(socio);
29     }
30
31
32     @Override
33     public Socio buscarPorId(Long id) {
34         // TODO Auto-generated method stub
35         return _socioDao.findById(id).get();
36     }
37
38     @Override
39     public void eliminarSocio(Socio socio) {
40         // TODO Auto-generated method stub
41         _socioDao.delete(socio);
42     }
43
44
45     @Override
46     public List<Socio> listarSocios() {
47         // TODO Auto-generated method stub
48         return (List<Socio>) _socioDao.findAll();
49     }
50 }
```

- findByEmail(String email) : Socio - SocioDao
- findBySocioPorDni(String socioDni) : Socio - SocioDao
- save(S entity) : S - CrudRepository
- count() : long - CrudRepository
- equals(Object obj) : boolean - Object
- existsById(Long id) : boolean - CrudRepository
- findAll() : Iterable<Socio> - CrudRepository
- findAllById(Iterable<Long> ids) : Iterable<Socio> - CrudRepository
- findById(Long id) : Optional<Socio> - CrudRepository
- findBySocioVotante() : List<Socio> - SocioDao
- getClass() : Class<?> - Object
- hashCode() : int - Object
- saveAll(Iterable<S> entities) : Iterable<S> - CrudRepository
- toString() : String - Object
- delete(Socio entity) : void - CrudRepository
- deleteAll() : void - CrudRepository
- deleteAll(Iterable<? extends Socio> entities) : void - CrudRepository
- deleteById(Long id) : void - CrudRepository
- notify() : void - Object

Fuente: Elaboración propia

En la figura 21 se puede apreciar que se está realizando una inyección a la interface SocioDao mediante la anotación @Autowired para poder consumir los metodos del CrudRepository, se

puede apreciar metodo llamado buscarPorDni se le esta pasando un parametro y mediante la inyeccion podemos ver todos los metodos que podemos utilizar sin haber realizado ninguna programación de CRUD o algun otro método en la capa DAO.

2.3.3.6. Recursos de publicación

2.3.3.6.1. Servidor privado virtual (VPS)

Una VPS es un servidor físico particionado en diferentes servidores virtuales que están siempre conectado al internet, cada servidor virtual es un espacio privado que puede ser configurado según la necesidad de cada empresa o usuario contratante. Para empezar a utilizar una VPS, tenemos que tener una planificaron de los diversos softwares que se van a instalar, así como el sistema operativo, firewall, servidor de aplicaciones, antivirus, etc.

Una VPS ofrece grandes ventajas que se consideran antes de adquirirla,

2.3.3.6.2. Proveedores de VPS

Algunas empresas que ofrece el servicio de alquiler de VPS se señalan a continuación:

- Amazon
- Digital Ocean
- Linode

La diferencia entre una y otro empresa mencionada puede variar se acuerdo al costo y a la complejidad de su configuración de cada uno de ellos. En este proyecto usaremos la VPS de Linode, por su fácil configuración, costo moderado y flexible además de ofrece protección sobre los ataques DDOS de manera gratuita a sus clientes.

2.3.3.6.3. *Servidor web Apache Tomcat*

El Apache Tomcat ® software es una implementación de código abierto del servlet de Java, Java Server Pages, Lengua y Java WebSocket tecnologías de expresión de Java. Las especificaciones de Java Servlet, JavaServer Pages, Java Expression Language y Java WebSocket se desarrollan bajo el Proceso de la comunidad Java (Apache Tomcat, s.f.).

Podemos decir que Apache Tomcat es un contenedor de servlets que se puede utilizar para publicar aplicaciones web realizada en java. Para que esta publicación web sea publicada se tiene que empaquetar todos los archivos que componen la aplicación web en un WAR (Archivo de aplicaciones web) e importarlo a Apache Tomcat.

Es importante mencionar que se tiene que realizar la debida habilitación de permisos necesarios en los archivos de configuración de Apache Tomcat, estas configuraciones están basadas en habilitar permisos de administrador a su panel de control, habilitar los permisos de publicación, configuración de puertos, etc.

Gestor de Aplicaciones Web de Tomcat

El gestor de aplicaciones web de Tomcat es un panel de control al que se puede acceder mediante un usuario y una contraseña que previamente han sido configurados, este panel de control permite visualizar información de la configuración y botones de acciones, a continuación, se muestra una figura que la describe.

Figura22

Panel de control de Tomcat

Aplicaciones					
Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/docs	Ninguno especificado	Tomcat Documentation	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/examples	Ninguno especificado	Servlet and JSP Examples	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/manager	Ninguno especificado	Tomcat Manager Application	true	1	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos

Desplegar	
Desplegar directorio o archivo WAR localizado en servidor	
Trayectoria de Contexto (opcional):	<input type="text"/>
Version (for parallel deployment):	<input type="text"/>
URL de archivo de Configuración XML:	<input type="text"/>
URL de WAR o Directorio:	<input type="text"/>
Desplegar	
Archivo WAR a desplegar	
Seleccione archivo WAR a cargar	<input type="button" value="Seleccionar archivo"/> <input type="text" value="Ningún archi..."/> <input type="button" value="seleccionado"/>
Desplegar	

Fuente: Elaboración propia

En la figura 22 se muestra la ruta donde de las aplicaciones, el nombre que recibe las aplicaciones, indica si la aplicación se está ejecutando, muestra el número de sesiones conectadas a esa aplicación. Además, nos muestra botones de acciones para parar, recargar o replegar la aplicación y en la parte inferior nos muestra el botón para poder seleccionar algún archivo WAR y poder publicarlo.

2.3.4. Seguridad

Es la capa de protección que se brinda al proyecto web para entornos backend como frontend utilizando herramientas tecnológicas apropiadas que brinden protección a la información almacenada en bases de datos de vulnerabilidades y posibles ataques mal intencionados o no autorizados.

2.3.4.1. Spring security

Cuando construimos una aplicación con tecnología web, está básicamente puede dividirse en dos apartados diferentes de desarrollo en un mismo proyecto como es el backend y el frontend, las fusiones de estos apartados nos brindan un producto final como es el software, sin embargo, el nivel de vulnerabilidad podría ser gigantesco si a los apartados mencionados no se le inyecta el nivel de seguridad requerido. Spring security es un framework de código abierto que nos va ayudar a mitigar esas vulnerabilidades y además se acopla fácilmente a plataformas desarrolladas con spring framework.

Spring Security es un marco que proporciona autenticación, autorización y protección contra ataques comunes. Con soporte para marcos de spring para aplicaciones imperativas y reactivas, es el estándar de facto para asegurar aplicaciones basadas en Spring, Spring Security requiere un entorno de ejecución Java 8 o superior. Tiene como objetivo operar de manera autónoma, no necesita colocar ningún archivo de configuración especial en su Java Runtime Environment. En particular, no es necesario configurar un archivo de política especial del Servicio de autenticación y autorización de Java (JAAS) ni colocar Spring Security en ubicaciones de rutas de clases comunes (Spring Security, 2020).

Podemos decir que Spring Security es un marco de trabajo que se integra de manera simple a cualquier proyecto realizado con spring, contiene funciones de seguridad para protegernos de ataques informáticos y brinda protección contra vulnerabilidades.

Ámbito De Seguridad

Spring Security nos permite cubrir diferentes ámbitos muy importantes de la aplicación web de manera simple y segura, aplicando diferentes filtros de seguridad según el espacio de desarrollo, estos ámbitos son:

- **La Autenticación**

Este ámbito permite verificar e identificar a quien pretende acceder a nuestra aplicación y una de las formas de autenticar a los usuarios que intentan ingresar es solicitar un usuario y una contraseña que se tendrán que validar con la data ya registrada.

Spring Security proporciona un soporte integral para la autenticación. La autenticación es cómo verificamos la identidad de quién intenta acceder a un recurso en particular. Una forma común de autenticar a los usuarios es solicitar al usuario que ingrese un nombre de usuario y una contraseña. Una vez realizada la autenticación, conocemos la identidad y podemos realizar la autorización (Spring security, s.f.).

Cuando trabajamos con spring security en nuestra aplicación, tenemos que utilizar una interfaz ya configurada por spring security llamada PasswordEncoder, se utiliza para realizar una transformación unidireccional de contraseñas, esta interfaz es inyectada al algoritmo de encriptación que utilizemos para grabar la contraseña en nuestra base de datos, la misma que se graba bajo un patrón de encriptación, está encriptación es de manera unidireccional es decir solo se puede cifrar y no descifrar, para el descifrado se utilizan mecanismos de recuperación mediante datos proporcionados por el en usuario, generalmente utilizamos el email personal de usuario.

Algoritmos de encriptación

La función de principal de un algoritmo de encriptación es hacer el cifrado de la información que se le envié previa autenticación, la complejidad, robustez y niveles de seguridad de cada algoritmo podrían ser diferentes.

La documentación de spring security en su versión 5.4.2 nos brinda información sobre algunos algoritmos de encriptación que se muestran a continuación:

- DelegarPasswordEncoder
- BCryptPasswordEncoder
- Argon2PasswordEncoder
- Pbkdf2PasswordEncoder
- SCryptPasswordEncoder

• La Autorización

La autorización es una etapa después de haber superado la autenticación, permite direccionar al usuario ingresante a ciertos métodos o páginas de nuestra aplicación, para ello se validará su rol asignado.

Rol de Seguridad

Podemos definir un rol como el nivel de permiso asignado que tiene el usuario que interactúa con nuestra aplicación, generalmente estos roles se agrupan por categorías de permisos tales como:

Rol Administrador: Generalmente tiene todos los permisos asignados, no se le limita con ninguna restricción.

Rol de Usuario: Dependiendo del análisis previo, podemos determinar a que URL o métodos puede tener acceso este usuario, generalmente tienen privilegios de navegación limitados o personalizados.

Es importante mencionar que cuando trabajamos con roles, debemos de tener en cuenta aquellas URL que tienen que ser de acceso público tales como el formulario de login o formulario de recuperación de clave, para ello se tiene que realizar la configuración de estas URL para sean de acceso público.

- **Protección contra vulnerabilidades**

Algunas de las protecciones que spring security nos ofrece contra vulnerabilidades se mencionan a continuación.

Cross Site Request Forgery (CSRF)

Es un tipo de ataque que busca la falsificación mediante solicitudes de peticiones web, inician con buscar la vulnerabilidad de capturar la sesión iniciada del usuario para llevarlo a otra página maligna con fallada amigable con los datos del usuario atacado y realizar diferentes operaciones o transacciones sin su consentimiento.

Spring security de manera automática nos ayuda a protegernos de este ataque mediante el patrón de **token** sincronizados automáticos que son generados por cada petición en nuestro servidor, es decir si un atacante captura la sesión del usuario tendrá que capturar también el token que genera spring security en nuestro servidor, y si decide realizar una operación maliciosa este fallara por que el token se genera de manera automática aleatoriamente en cada petición de nuestra aplicación.

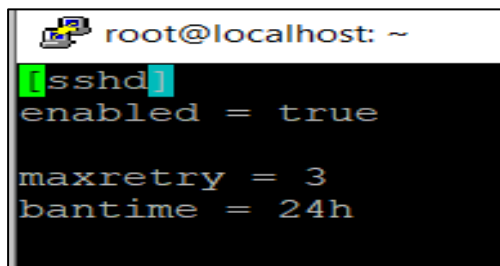
La razón por la que es posible un ataque CSRF es que la solicitud HTTP del sitio web de la víctima y la solicitud del sitio web del atacante son exactamente iguales. Esto significa que no hay forma de rechazar solicitudes provenientes del sitio web maligno y permitir solicitudes provenientes del sitio web del banco. Para protegernos contra ataques CSRF, debemos asegurarnos de que hay algo en la solicitud que el sitio maligno no puede proporcionar para que podamos diferenciar las dos solicitudes (Spring, s.f.).

Seguridad de consola con fail2ban

Prevenir ataques a nivel de Linux, actúa con firewall o cortafuego para bloquear Ip desconocidas que quieran vulnerar la seguridad con intentos fallidos varias veces, para ello esta aplicación las detecta y las encierra por el tiempo de configuración que se le haya establecido, se pueden establecer diferentes configuraciones hasta el aviso mediante un correo al administrador del sistema. (Barrientos, 2021).

Figura23

Configuración fail2ban - seguridad de consola



```
root@localhost: ~  
[sshd]  
enabled = true  
maxretry = 3  
bantime = 24h
```

Elaboración propia

Nota. La figura 23 muestra la configuración que se ha establecido, en ese caso el comando enabled en true, permite activar el servicio maxretry permite programar el máximo de intentos fallidos, es decir si el intruso falla más de tres veces será encarcelado por 24 horas como lo dice el comando bantime.

Una vez que esa Ip fueron baneadas el firewall las mantendrá encerradas como se muestra en la siguiente figura.

Figura24

Ip baneadas por intento de ataque

```
You have new mail.
Last login: Thu Dec  9 12:10:14 2021 from 190.233.26.132
root@localhost:~# sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 2
|  |- Total failed:    69865
|  `-- File list:      /var/log/auth.log
`-- Actions
    |- Currently banned: 1
    |- Total banned:    17663
    `-- Banned IP list: 195.29.102.21
root@localhost:~#
```

Elaboración propia

Nota. La figura 24 muestra que se han baneado o bloqueado 69865 Ip que han intentado vulnerar nuestro sistema a través de ataques por consola.

Redirección HTTP

Spring security te permite configurar de manera simple el re-direccionamiento a páginas web seguras, es decir se podría configurar para que toda la navegación de la aplicación con HTTP pueda ser enviada de manera automática al protocolo HTTPS.

HTTP Firewall

Spring security se puede configurar su propio firewall, registrando listas blancas que contengan los métodos y url de nuestra aplicación para que puedan ser reconocidas, también se pueden listas negras de caracteres para que el firewall los rechace en el intento de navegación.

Limitar sesiones

Spring security nos permite limitar de manera sencilla la cantidad de sesiones activa que puede tener cada usuario, es decir si el usuario ya inicio sesión, nadie más podrá hacerlo hasta que cierre o caduque la sesión.

Nivel De Seguridad

Spring Security aplica seguridad en dos niveles de este proyecto, estos niveles son:

- **A nivel de peticiones web (http request)**

Este nivel de seguridad permite restringir el acceso a peticiones de páginas de nuestro proyecto. Para ello se estable la configuración de autorizaciones aquellas rutas que serán de uso público y privado y solo podrán acceder según su rol asignado del usuario.

Figura 25

Seguridad a nivel de Url - Http request

```
@EnableWebSecurity
public class WebSecurityConfig extends WebSecurityConfigurerAdapter{

    @Override
    protected void configure(HttpSecurity http) throws Exception {

        http.authorizeRequests()
            .antMatchers("/index",
                "/home",
                "/",
                "/css/**",
                "/bcrypt/**",
                "/js/**",
                "/images/**",
                "/forgot-password**", // 6
                "/reset-password**").permitAll() // 7
            .antMatchers("/views/socio/**").hasAnyRole("ADMIN")

            .antMatchers("/views/socios/listar").hasAnyRole("USER")
            .antMatchers("/views/socios/save").hasAnyRole("ADMIN")
    }
}
```

Fuente: Elaboración propia

- **A nivel de métodos**

Este nivel de seguridad es aplicado a nuestros métodos de la capa controladora, es un nivel de seguridad mucho más profesional y específico. Podemos establecer permisos a toda la clase controladora o solo ciertos métodos o compartir los permisos dependiendo del rol asignado.

Figura26

Seguridad a Nivel de Métodos

```
@RequestMapping("/listar")
@Secured({"ROLE_ADMIN"})
public String listarPropuesta(Model model) {
    List<Propuesta>propuesta=_propuestaService.listarPropuesta();
    model.addAttribute("propuestasListar", propuesta);
    return "/views/propuesta/listaPropuesta";
}

@Secured({"ROLE_USER", "ROLE_ADMIN"})
@RequestMapping("/listarResultado")
public String listarPropuestaResultado(Model model) {

    List<Propuesta>propuesta=_propuestaService.listarPropuesta();
    List<Cargo>cargosListar=_cargoService.findByCargoAll();

    model.addAttribute("propuestasListarResultado_x", propuesta);
    model.addAttribute("listarCargoResultado_x", cargosListar);

    return "/views/propuesta/listaPropuestaResultado";
}
```

Fuente: Elaboración propia

Es importante mencionar que antes de la autenticación se tiene que realizar el registro de los usuarios, Spring Security nos brinda un estándar para la tabla de usuario donde se tiene que tener ciertos campos ya definidos como es **el nombre de usuario, estado del usuario, rol del usuario y contraseña**. De esta manera Spring Security se encarga de validar esos campos para su autenticación y permiso respectivo.

2.3.4.2. Security sockets layer (SSL)

Es un protocolo de seguridad que se encarga de hacer la encriptación de los datos que viajan de nuestro servidor web y algún navegador web, otorgando seguridad al usuario que se encuentra navegando en la página web. La correcta configuración de este protocolo hace que nuestra página

web tenga como predominación HTTPS, algunos navegadores web que no utilizan esta predominación muestran una alerta de página insegura, perdiendo credibilidad ante los usuarios.

Tipos de certificado SSL

Existen diferentes tipos de certificados SSL para cada escenario como se muestran a continuación:

SSL por dominio: Para adquirir este tipo de certificado se tiene que demostrar ante la empresa certificadora que el dominio te pertenece.

SSL por organización: Este tipo de certificado es otorgados para empresas total mente constituida, se tienen que proveer datos de la empresa para su validación y verificación.

SSL por Multidominios: Es un tipo de certificado que permite asegurar varios nombres de dominio.

Los precios de cada uno de los certificados pueden varias de acuerdo a las promociones u ofertas de cada empresa certificadora.

SSL de licencias gratis

Existen empresa que nos brindan certificado SSL, generalmente te limitan en tiempo y numero de dominio, es decir se tiene que renovar cada cierto tiempo y generalmente solo brindan seguridad a un dominio mas no a subdominio. Sin embargo, si utilizamos Spring boot solo necesitamos SSL para un dominio y las demás paginas las podemos vincular por la ruta de cada método, a continuación de mencionan dos empresas que brindad este servicio.

- **Let's Encrypt**

Let's Encrypt es una autoridad de certificación gratuita, automatizada, y abierta traída a ustedes por la organización sin ánimos de lucro Internet Security Research Group (ISRG) (Let's Encrypt, s.f.).

- **ZeroSSL**

Proteja sus sitios web con certificados SSL de 90 días sin cargo registrándose en el plan gratuito ZeroSSL. La validación del certificado toma solo unos minutos y la instalación es tan fácil como es posible con instrucciones en cada paso del camino (ZeroSSL, s.f.).

2.3.4.3. Criptografía

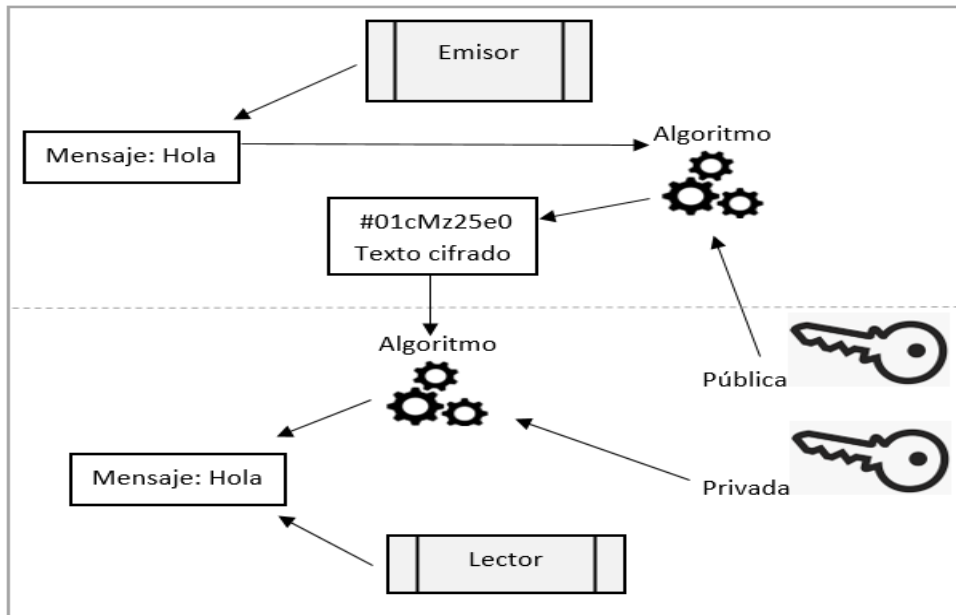
En cumplimiento de la Constitución Política del Perú vigente en su **Artículo 31**, en unos de sus párrafos indica lo siguiente, “El voto es personal, igual, libre y **secreto**”. Para dar cumplimiento a la ley mencionada necesitamos de argumentos tecnológicos que mantenga el voto de manera secreta, es allí donde la criptografía juega un rol muy importante.

2.3.4.3.1. Criptografía Asimétrica

La criptografía asimétrica también se le conoce como cifrado de clave pública, es un método de encriptación que requiere dos claves, una clave pública y una clave privada para poder funcionar. La clave pública puede ser vista por cualquier persona y la clave privada debe de mantenerse en secreto, es decir; cuando el emisor quiere enviar un mensaje por internet busca la clave pública del receptor y cifra su mensaje con esa clave, luego el receptor puede descifrar el mensaje con su llave privada. La complejidad de su descifrado radica en la complejidad de las funciones matemáticas que la componen y el alto requerimiento de computo que se requiere para el descifrado.

Figura27

Criptografía Asimétrica



Fuente: Elaboración propia

La figura 27 representa el uso del algoritmo asimétrico, donde el usuario para realizar envío de un mensaje utiliza la llave pública del lector para poder realizar el cifrado, una vez que el cifrado se ha realizado se realiza el envío al lector, el lector utiliza su llave privada para descifrar el mensaje y leer el contenido.

2.4. Definición de términos básicos

Beans

Son objetos que maneja el contenedor Spring, es una clase con una previa configuración para que el core container los pueda reconocer.

Framework

Podemos definir un framework aplicado a desarrollo web, es un marco de trabajo compuesto de conjunto de herramienta estandarizadas que agilizan, simplifican y evitan la redundancia en el desarrollo del software.

Bakend

Es el estereotipo que le damos al programador que se encarga del desarrollo del software del software y siempre está interactuando del lado del servidor, a modo de analogía sería como la persona que construye el motor de un vehículo.

Frontend

Es el estereotipo que le damos al programador que se encarga de la presentación del proyecto y siempre está interactuando con algún navegador, continuando con la analogía anterior, sería la persona que se encargará de buscarle un caparazón al motor y los colores que este debería tener.

Ddos

Por sus siglas Distributed Denial Of Service o ataque distribuido denegación de servicio, es un ataque informático que simulan muchas computadoras atacando al servidor con la finalidad de hacer caer el servicio web.

Sistema Operativo

Es un conjunto de programas para administrar los recursos de una computadora de los más conocidos, Linux y Windows.

Base de datos

Una base de datos permite almacenar información de manera ordena, juega un rol importante dentro del desarrollo web, actúa como nuestro repositorio de datos para ser invocada cuando la aplicación lo requiera.

Plugins

Los plugins son pequeños programas complementarios que amplían las funciones de aplicaciones web y programas de escritorio.

Api rest

Es una interfaz de aplicaciones para transferir datos, es un conjunto de subrutina, funciones y procedimientos creados para ser consumido por cualquier otro software.

Antivirus

Conjunto de programas que se encargan de proteger y eliminar virus de un computador.

Thymeleaf

Es un moderno motor de plantillas Java del lado del servidor para entornos web e independientes.

SQL injection

Es un ataque informático que busca vulnerabilidades en las consultas SQL para infiltrarse con intenciones maliciosas.

Token

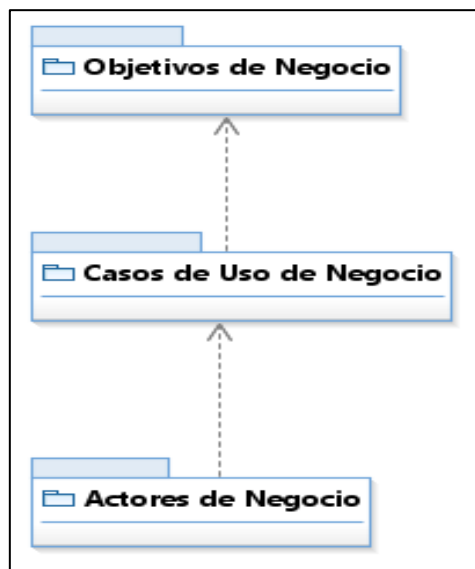
Es un recurso que se utiliza como una capa de seguridad para pasar a la siguiente etapa de algo, es un código encriptado que se genera para permitir realizar alguna acción en el momento y caduca cuando la acción fue realizada.

CAPITULO III. DESARROLLO DEL PRODUCTO

3.1. Modelo de casos de uso del negocio – MCUN (Vista externa)

Figura28

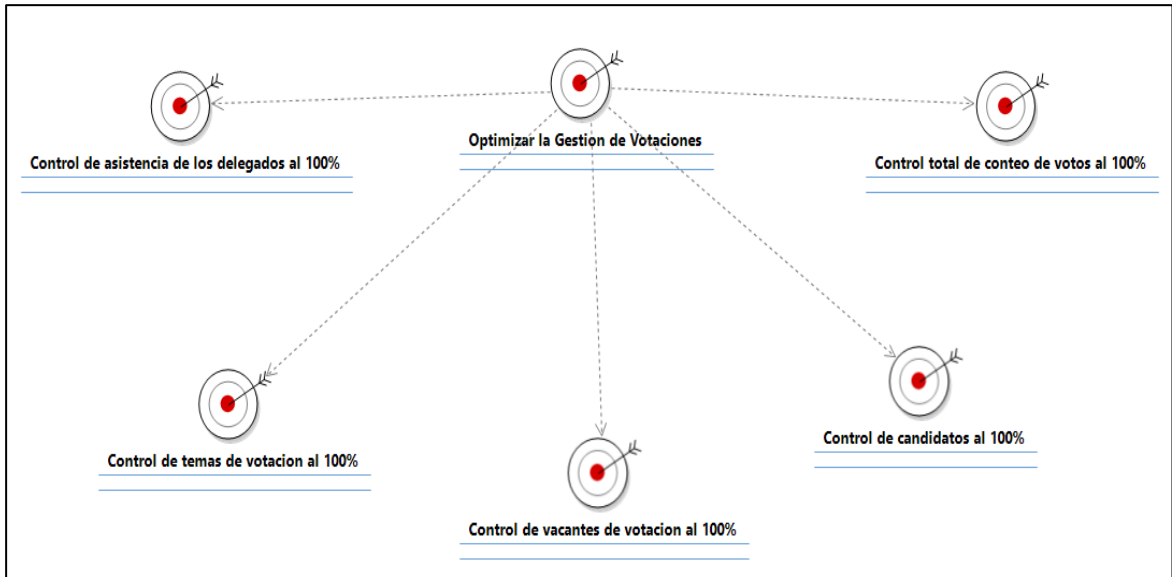
Estructura principal del MCUN



Fuente: Elaboración propia

Figura 29

Diagrama de objetivos del negocio



Fuente: Elaboración propia

Figura 30



Objetivos del negocio general - ON




Objetivo general	Descripción
 <p>Optimizar la Gestión de Votaciones</p>	<p>Tiene como finalidad la realización de las actividades de votaciones de la manera más eficiente posible.</p>

Fuente: Elaboración propia

Figura 31

Objetivo específico

Objetivo específico	Descripción
 <p><u>Control de asistencia de los delegados al 100%</u></p>	<p>El 100 % de los delegados presentes, representa aquellos delegados que pasaron la asistencia y podrán participar en las diferentes votaciones, por ello si un delegado se encuentra presente pero no emite su votación, ya sea por problemas técnicos o simplemente no desea votar, deberá aparecer en el grupo de delegados no votantes, para el conteo final.</p>
 <p><u>Control de temas de votacion al 100%</u></p>	<p>Los temas de votación son los puntos de agenda o acuerdos de asamblea que fueron llevados a votación, este control permitirá conocer quienes votaron a favor, en contra o se abstuvieron de votar.</p>


 <p>Control de vacantes de votacion al 100%</p>	<p>Los vacantes de votación son los cargos administrativos puestos en disputa mediante las votaciones, el o los delegados de mayor votación serán los designados a hacerse del cargo según número de votaciones.</p>
 <p>Control de candidatos al 100%</p>	<p>Tener el control de candidatos, implica conocer que delgado está candidateando para alguna vacante administrativa.</p>
 <p>Control total de conteo de votos al 100%</p>	<p>Tener el control del conteo de votos es parte fundamental de la transparencia de las votaciones, en el conteo de votos participan todos los votos de los delegados presentes que votaron y los presentes que no votaron.</p>

Fuente: Elaboración propia

Figura 32

Casos de uso de negocio

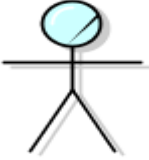
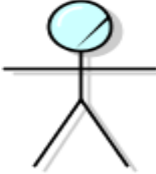
Casos de uso de negocio	Descripción
--------------------------------	--------------------

 Gestion de Votaciones	<p>El caso empieza cuando se tienen que celebrar asambleas o votaciones dentro de la empresa, el presidente da la orden a</p>
---	---

Fuente: Elaboración propia

Figura 33

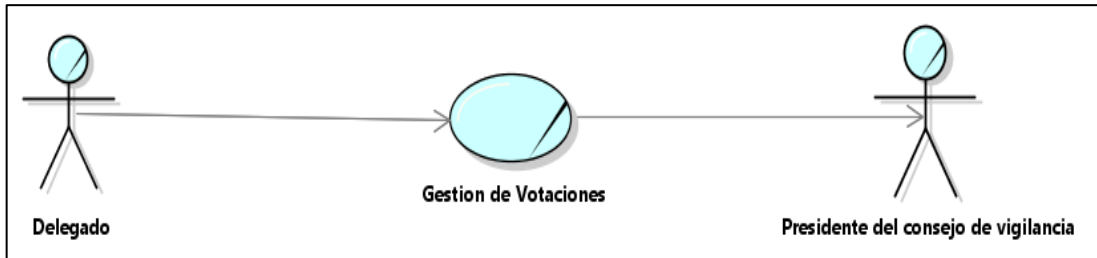
Actores de negocio

Actores de negocio	Descripción
 Delegado	<p>Es el usuario que tiene que realizar las votaciones por los temas o candidatos propuestos dentro de la asamblea, además un delegado también puede ser un candidato.</p>
 Presidente del consejo de vigilancia	<p>Es el encargado de verificar el conteo correcto de los votos realizados por los delegados.</p>

Fuente: Elaboración propia

Figura34

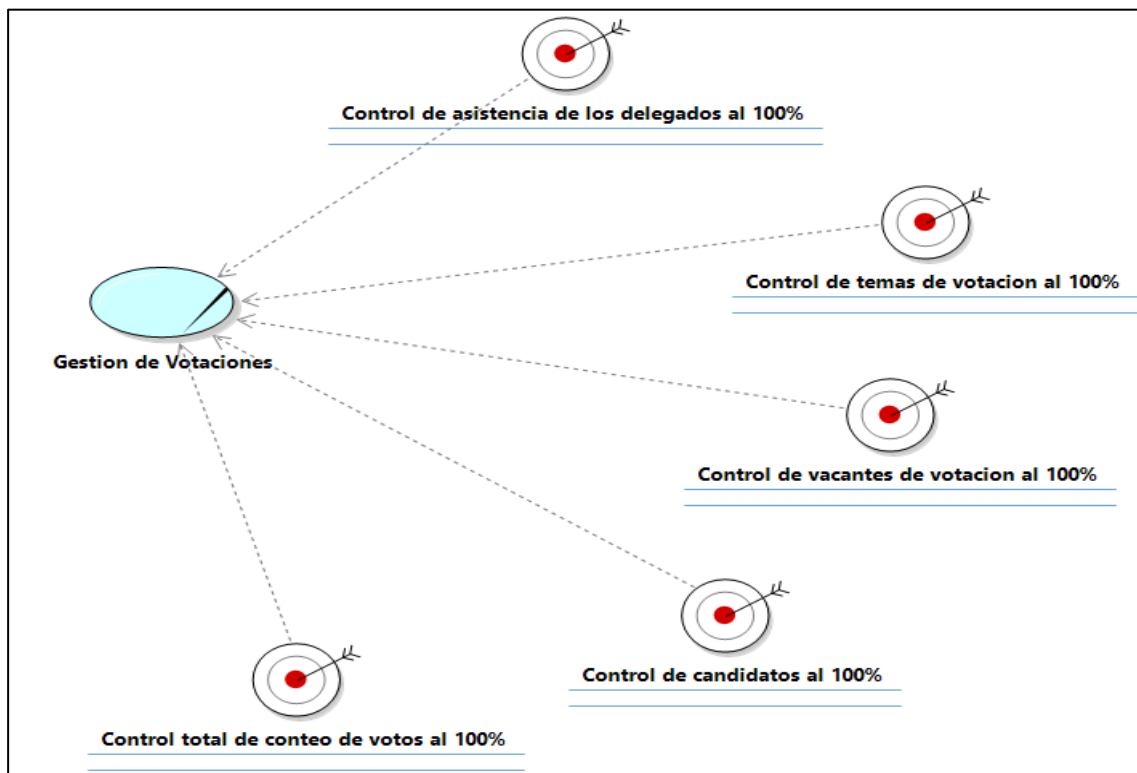
Diagrama general de casos de uso del negocio



Fuente: Elaboración propia

Figura 35

Diagrama de objetivos del negocio vs casos de uso del negocio

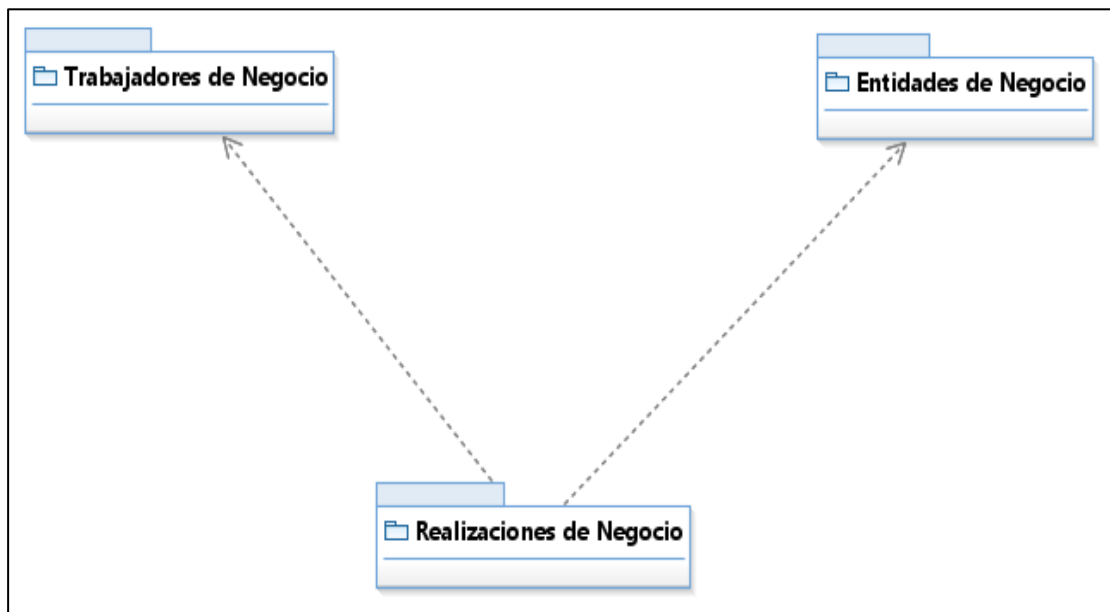


Fuente: Elaboración propia

3.2. Modelo de análisis del sistema (MAN) – vista interna

Figura 36

Organizar modelo de análisis del negocio






Fuente: Elaboración propia

Figura 37

Trabajadores de negocio

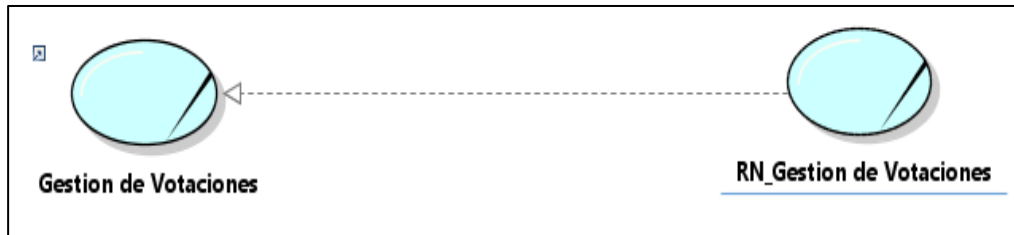
Trabajador del negocio	Descripción
------------------------	-------------

 <p><u>Secretaria</u></p>	<p>Es el persona con el rol de realizar el registro de los delegados que asistirán a la asamblea.</p>
 <p><u>Presidente del consejo de administración</u></p>	<p>Persona que se encarga de tomar la asistencia de los delegados inicialmente, dirige la asamblea para las votaciones de los diversos temas de agenda, da el inicio y fin de la votación para luego develar los resultados.</p>
 <p><u>Presidente del comité electoral</u></p>	<p>Persona que se encarga de tomas asistencia de los delegados antes de dar inicio a las votaciones de las diferentes vacantes, registra la vacante, los candidatos, da el inicio y fin de la votación para luego develar los resultados.</p>

Fuente: Elaboración propia

Figura 38



Realizaciones de negocio (RN)








Fuente: Elaboración propia

Figura 39

Entidades de negocio (EN)

Entidad del negocio	Descripción
 <p>Registro Delegado</p>	<p>Entidad representada por el delegado de la asamblea, cuenta con sus propiedades que la componen.</p>
 <p>Registro Asistencia</p>	<p>Entidad representada por la asistencia que se les toma a los delegados durante la asamblea y sus propiedades que la componen.</p>
	<p>Entidad representada por los temas o puntos de agenda que se llevan a votación durante la</p>

 <p>Registro Tema</p>	<p>asamblea y cuentan con sus propiedades.</p>
 <p>Registro Vacante</p>	<p>Entidad representada por las vacantes o cargos que se llevaran a votación durante la asamblea y cuentan con sus propiedades.</p>
 <p>Registro Candidato</p>	<p>Entidad representada por el candidato que postula a hacerse de algún cargo vacante durante la asamblea, cuenta con sus propias propiedades.</p>
 <p>Registro Voto Tema</p>	<p>Entidad representada por los votos realizados por los delegados referente a los temas o puntos de agenda durante la asamblea, esta entidad cuenta con sus propias propiedades.</p>

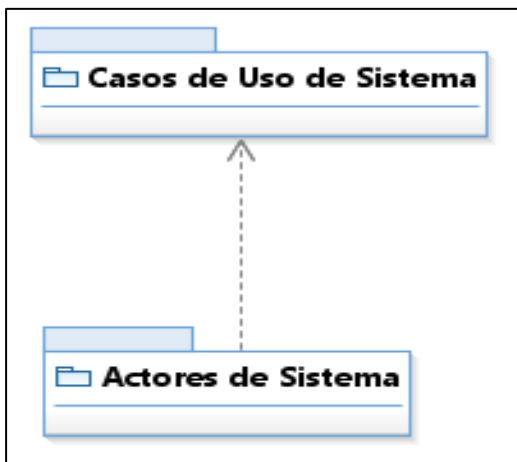
 <p style="text-align: center;">RegistroVotoCandidato</p>	<p>Entidad representada por los votos realizados por los delegados referente a los candidatos que postulan por vacantes o cargos administrativos durante la asamblea, esta entidad cuenta con sus propias propiedades.</p>
---	--

Fuente: Elaboración propia

3.3. Modelo de casos de uso del sistema (MCUS)

Figura 40

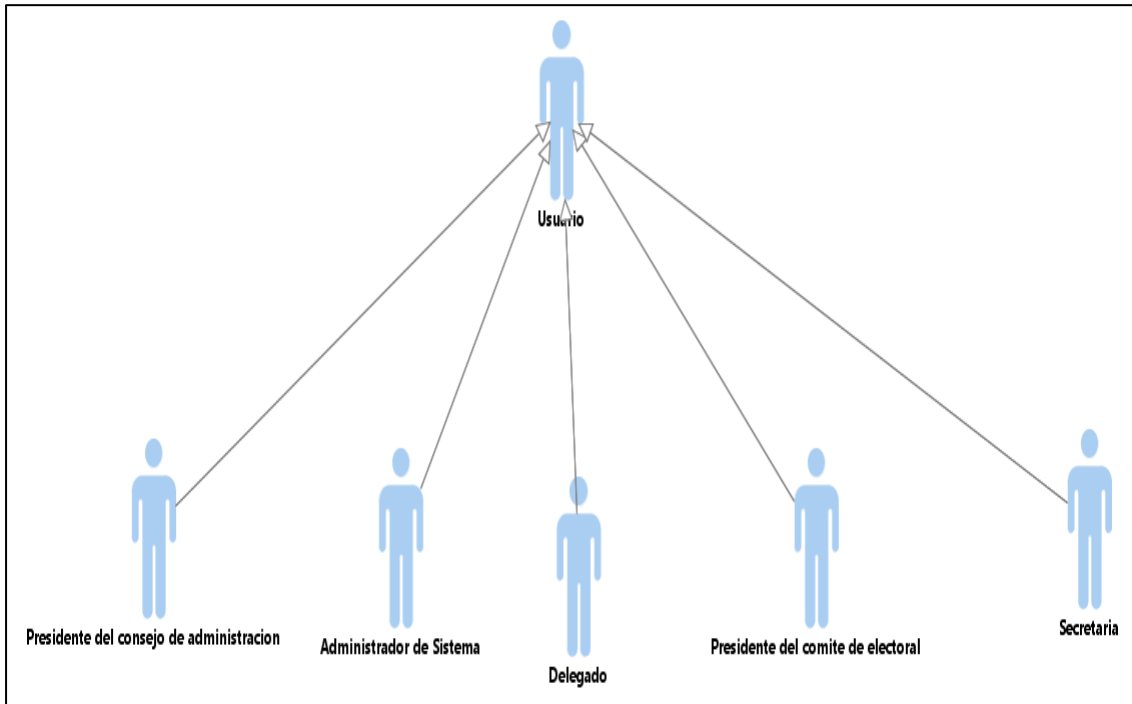
Organizar MCUS




Fuente: Elaboración propia





Figura 41

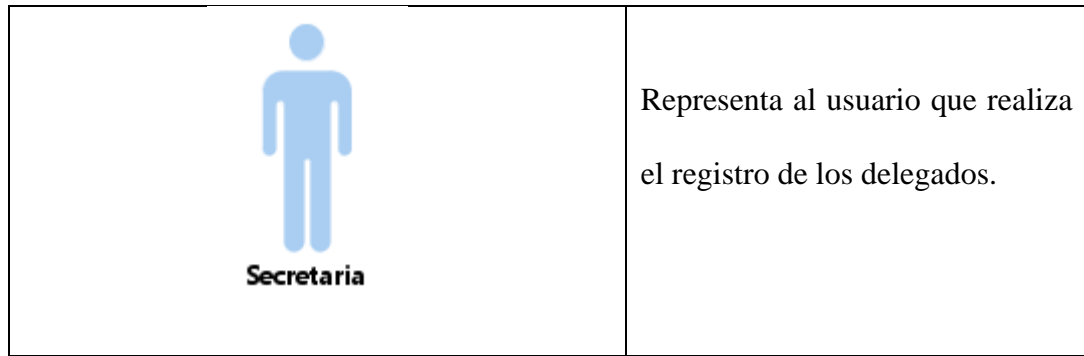
Actores del sistema



Fuente: Elaboración propia

Actores del sistema	Descripción
 Usuario	Representa a los usuarios del sistema que registran a los delegados, candidatos, vacantes, asistencia, etc.

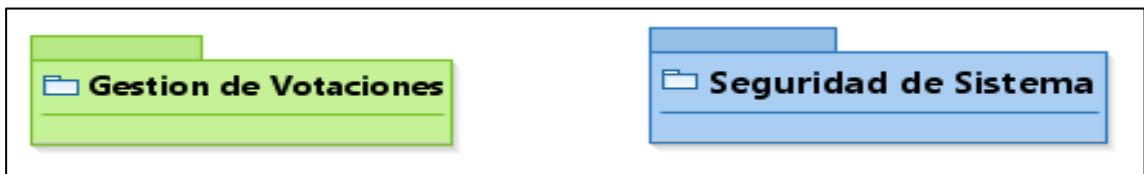
 <p>Administrador de Sistema</p>	<p>Representa al usuario con todos los privilegios del sistema.</p>
 <p>Presidente del consejo de administracion</p>	<p>Representa al usuario encargado de la toma de asistencia y registro de los temas o puntos agenda que se llevaran a votación, así como la lectura de los resultados del mismo.</p>
 <p>Delegado</p>	<p>Representa al usuario que participara en las diferentes votaciones dentro de la asamblea.</p>
 <p>Presidente del comite de electoral</p>	<p>Representa al usuario encargado de la toma de asistencia, registro de vacantes y candidatos, así como la lectura de los resultados.</p>



Fuente: Elaboración propia

Figura 42

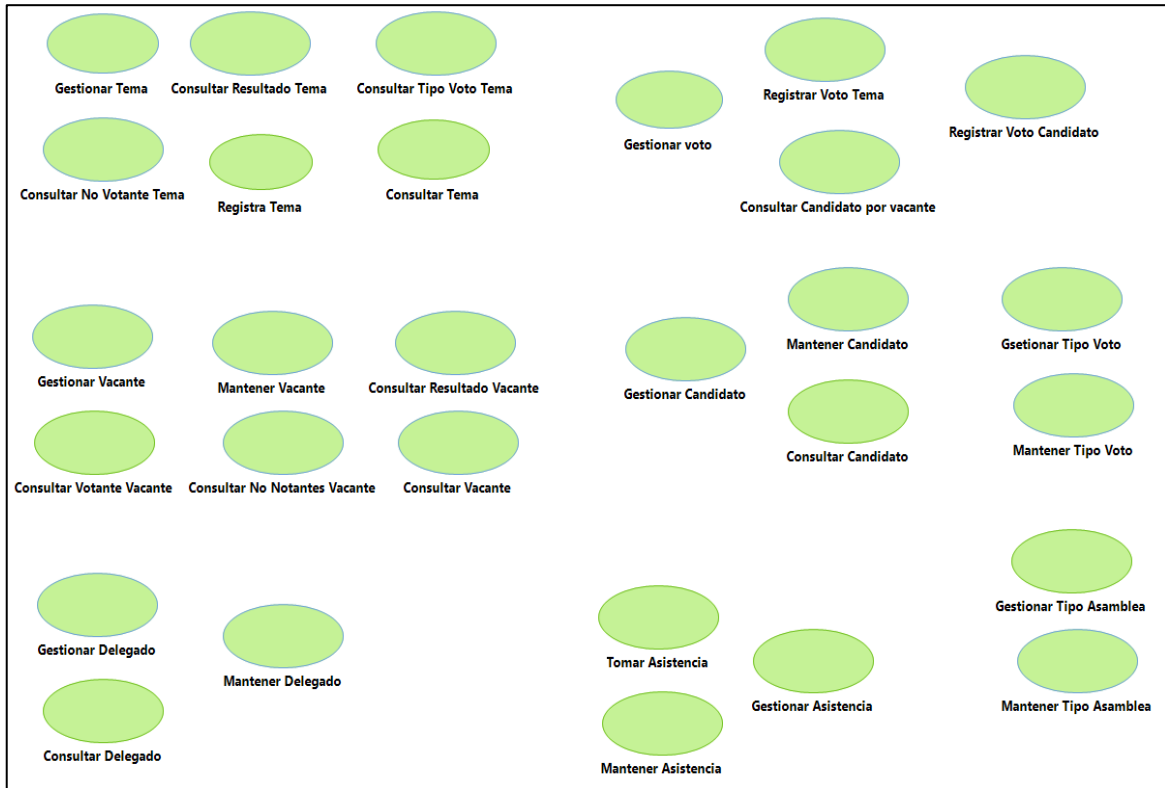
Paquetes de casos de uso del sistema



Fuente: Elaboración propia

Figura 43

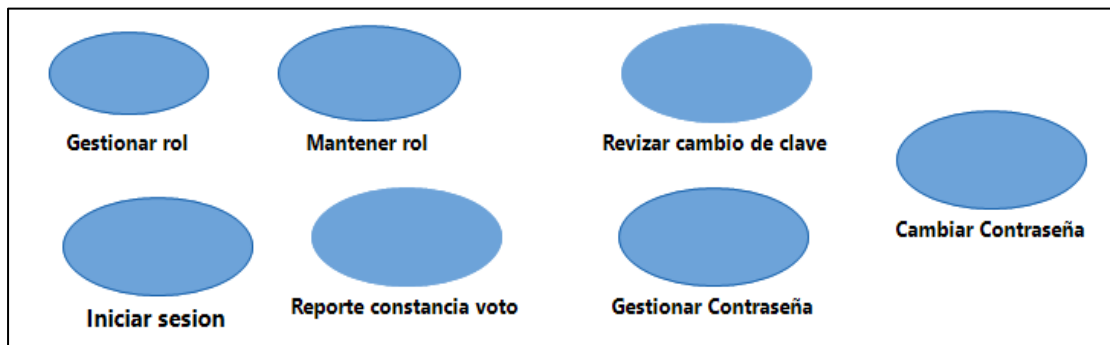
Casos de uso de Gestión de votaciones



Fuente: Elaboración propia

Figura 44

Casos de uso de Seguridad del sistema



Fuente: Elaboración propia

Figura 45

Diagrama general de CUS

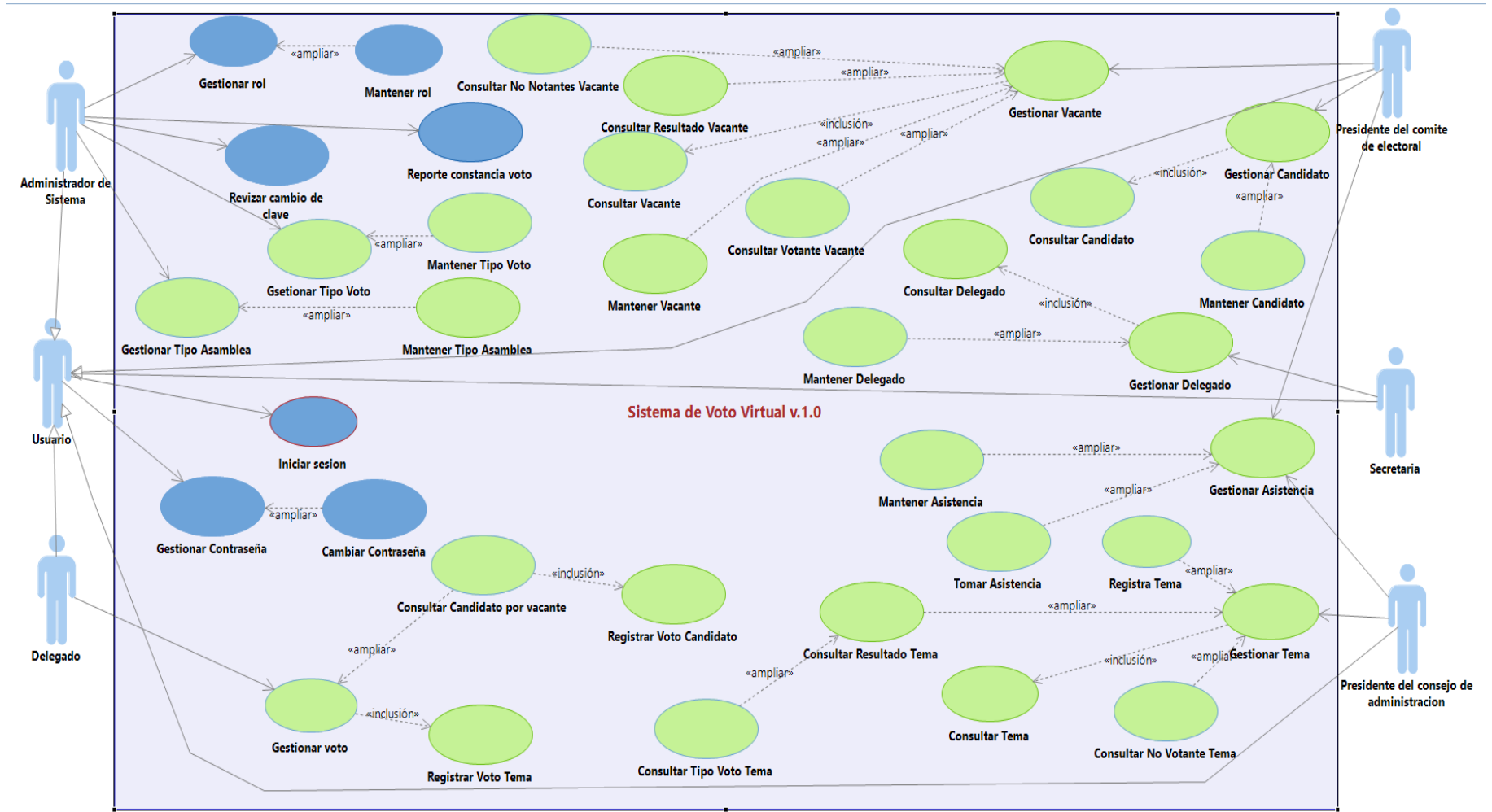
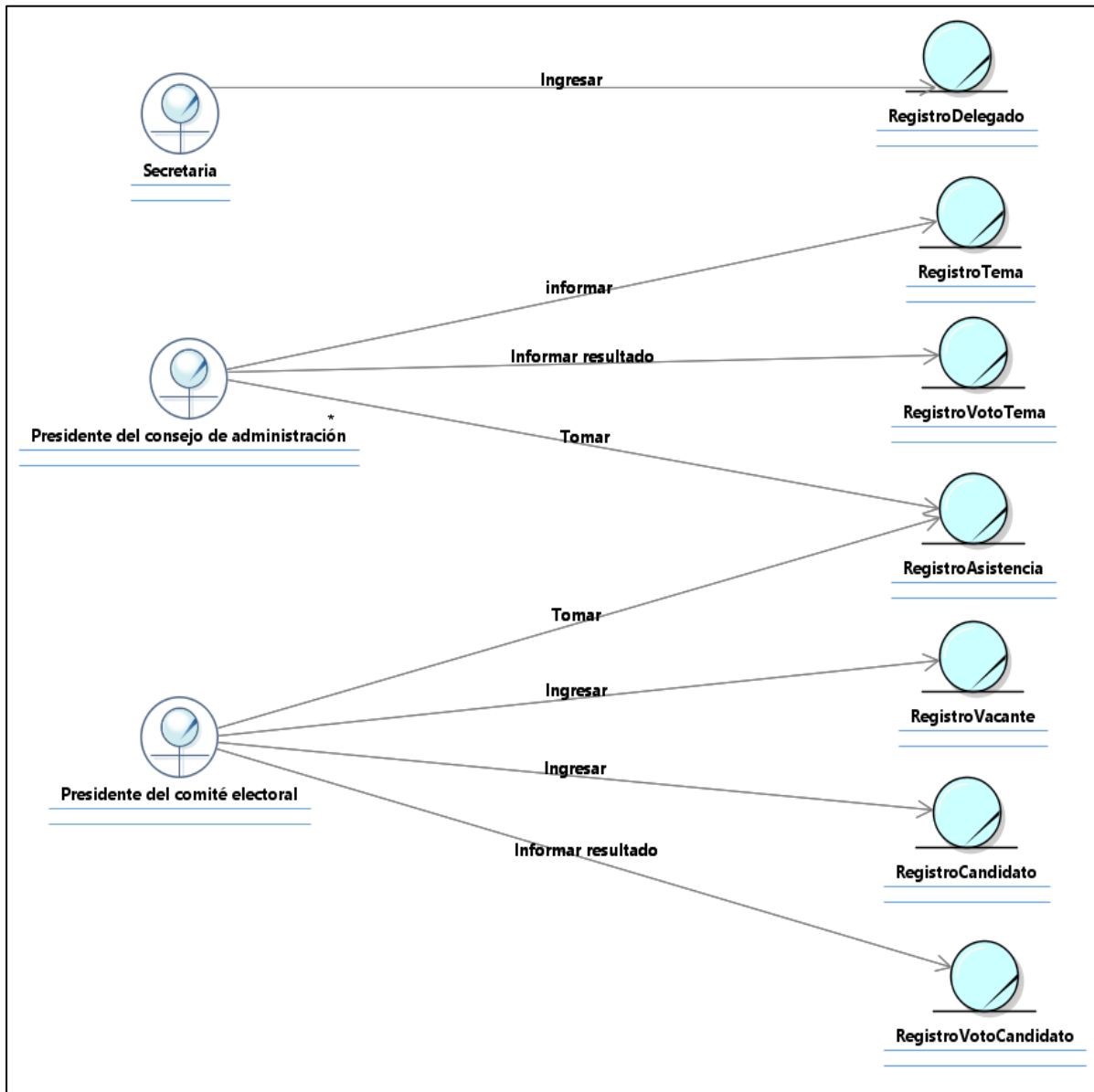


Figura46

Casos de usos del sistema

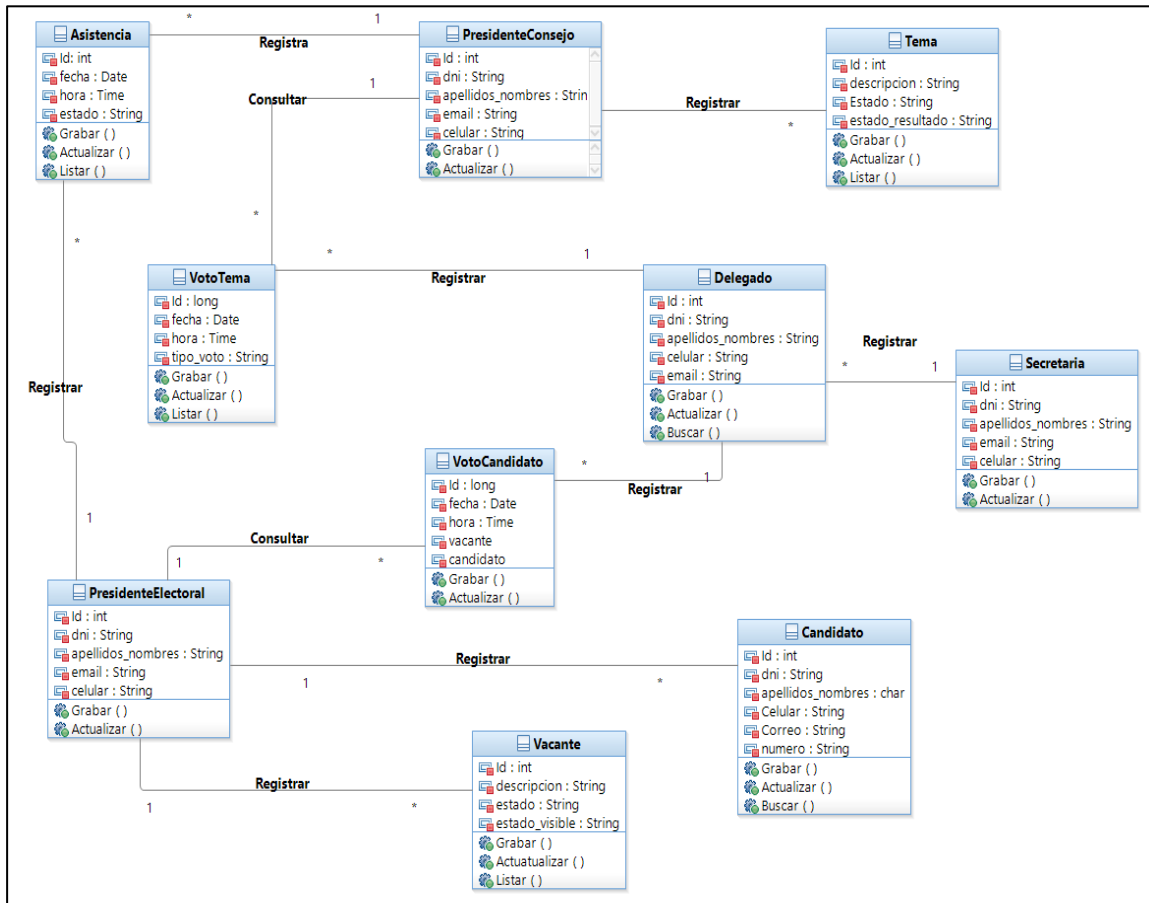


Fuente: Elaboración propia

3.4. Análisis y diseño del sistema (MCUS)

Figura 47

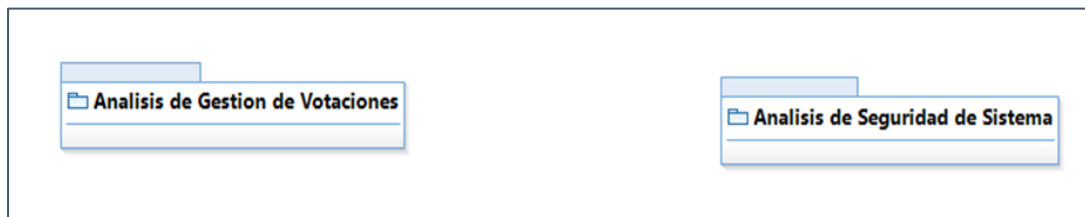
Diagrama de clases



Fuente: Elaboración propia

Figura 48

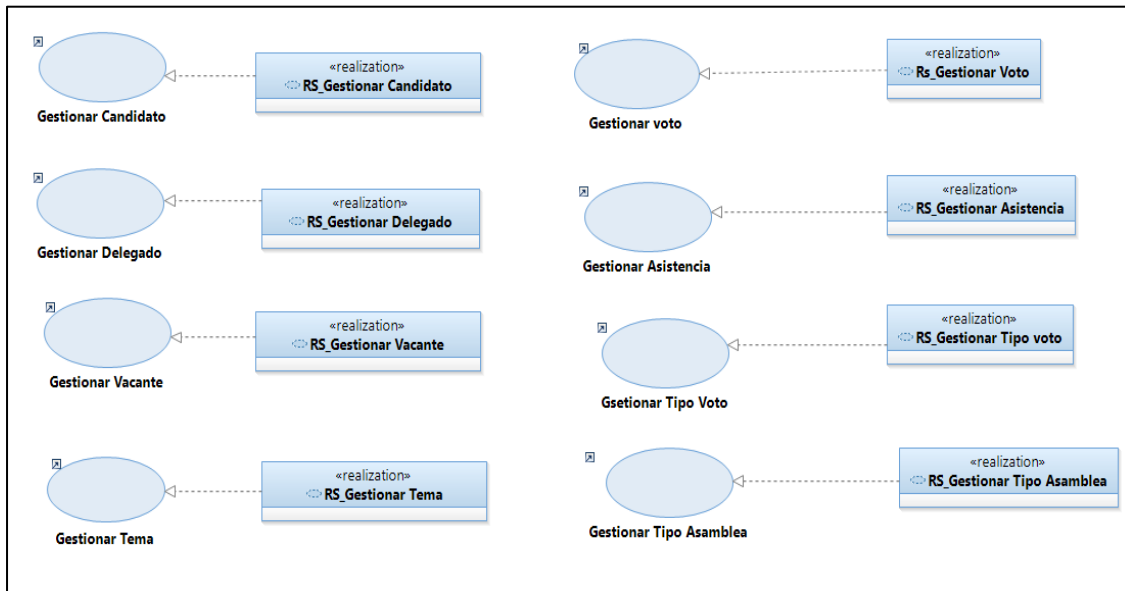
Organizar análisis y diseño



Fuente: Elaboración propia

Figura 49

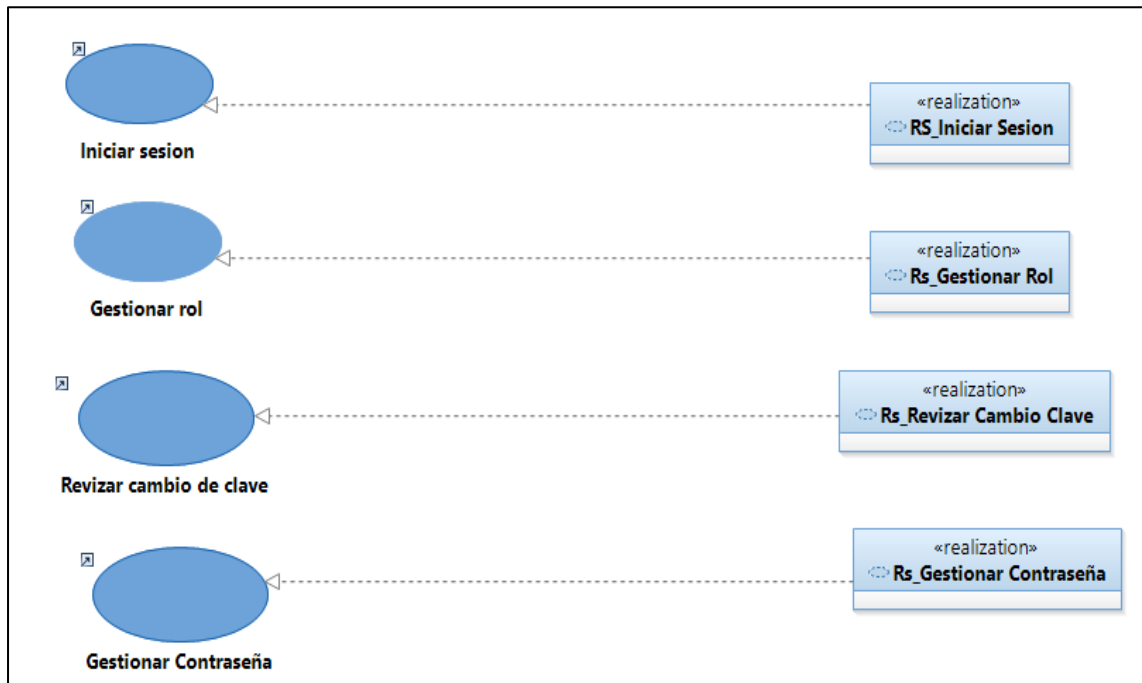
Análisis de gestión de votaciones



Fuente: Elaboración propia

Figura 50

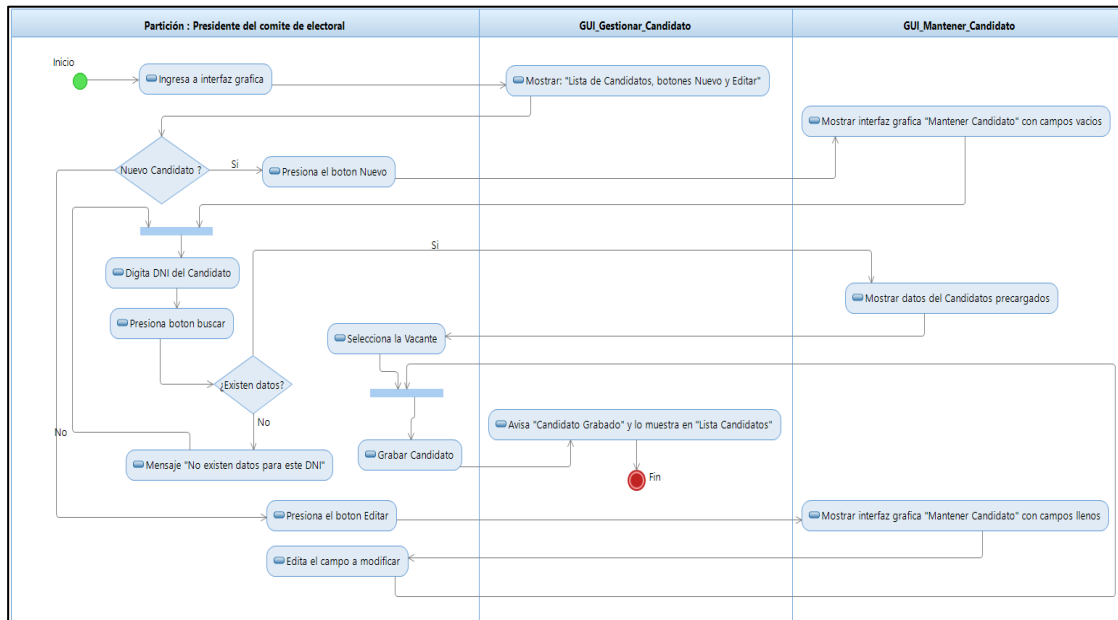
Análisis de seguridad del sistema



Fuente: Elaboración propia

Figura 51

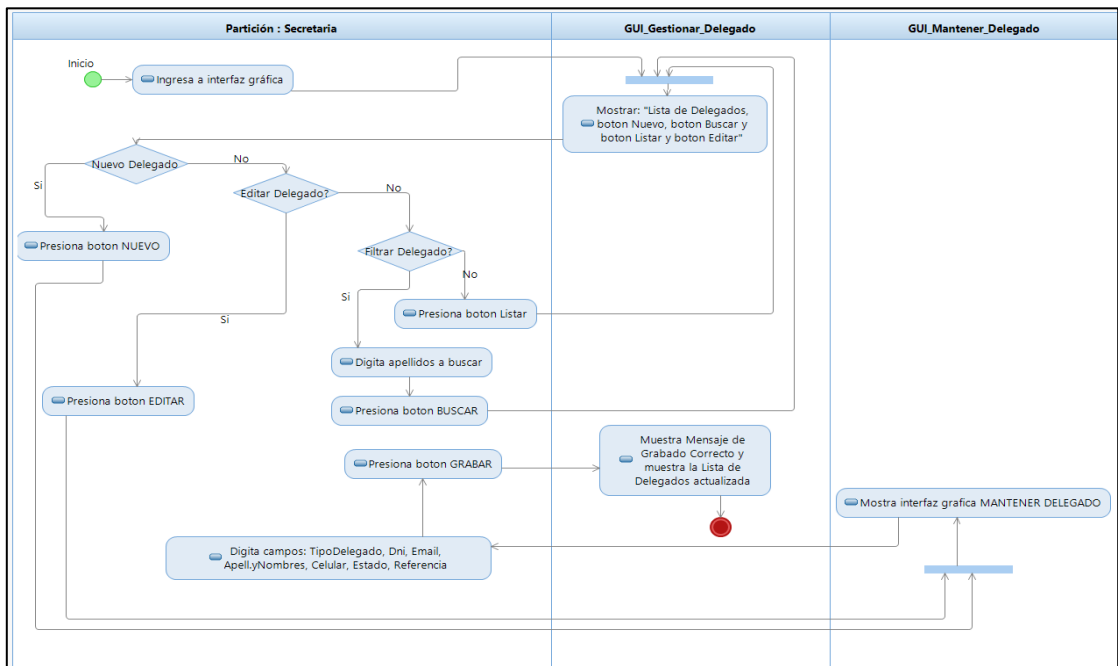
Diagrama de actividades – Gestionar Candidato



Fuente: Elaboración propia

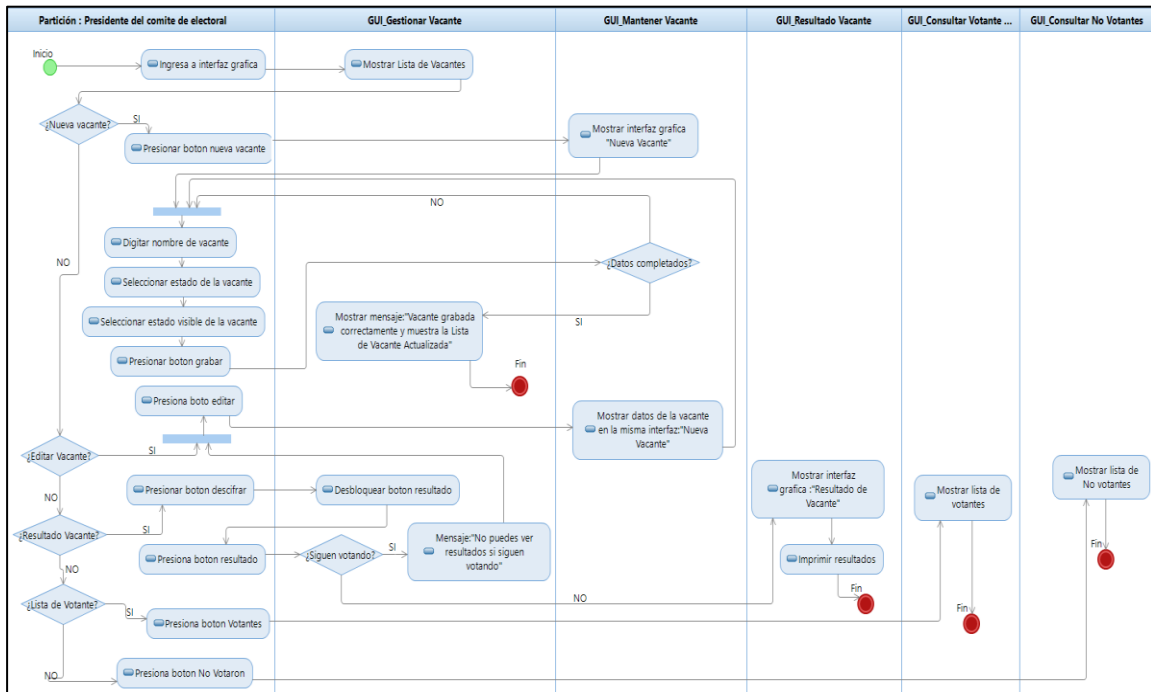
Figura 52

Diagrama de actividades – Gestionar Delegado



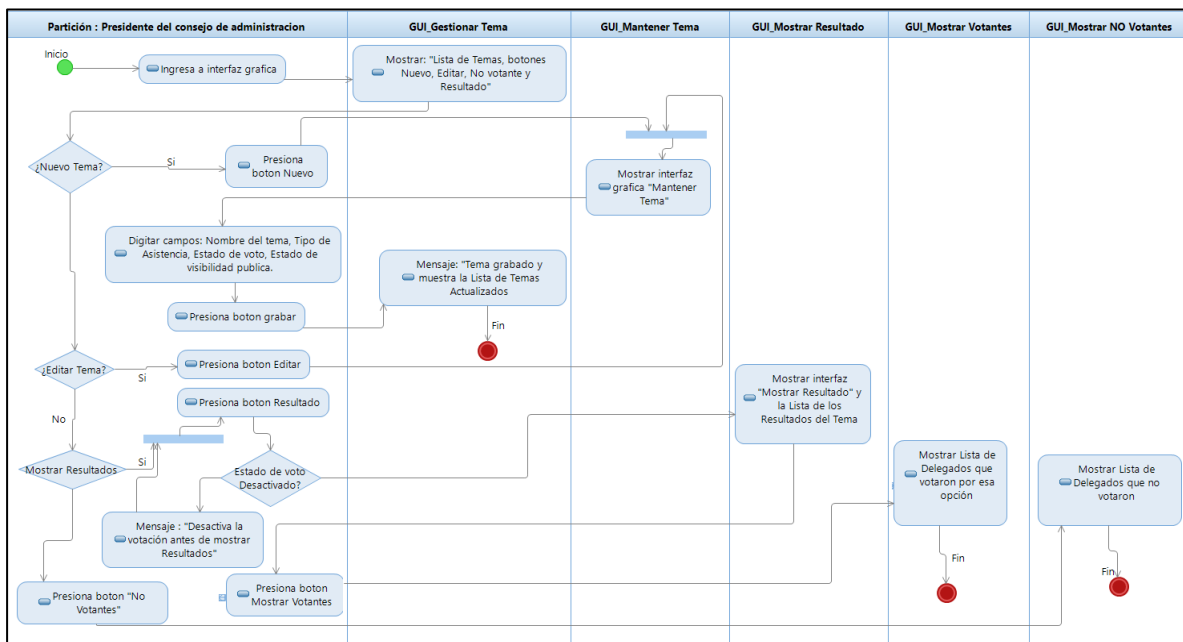
Fuente: Elaboración propia

Figura 53
Diagrama de actividades – Gestionar Vacante



Fuente: Elaboración propia

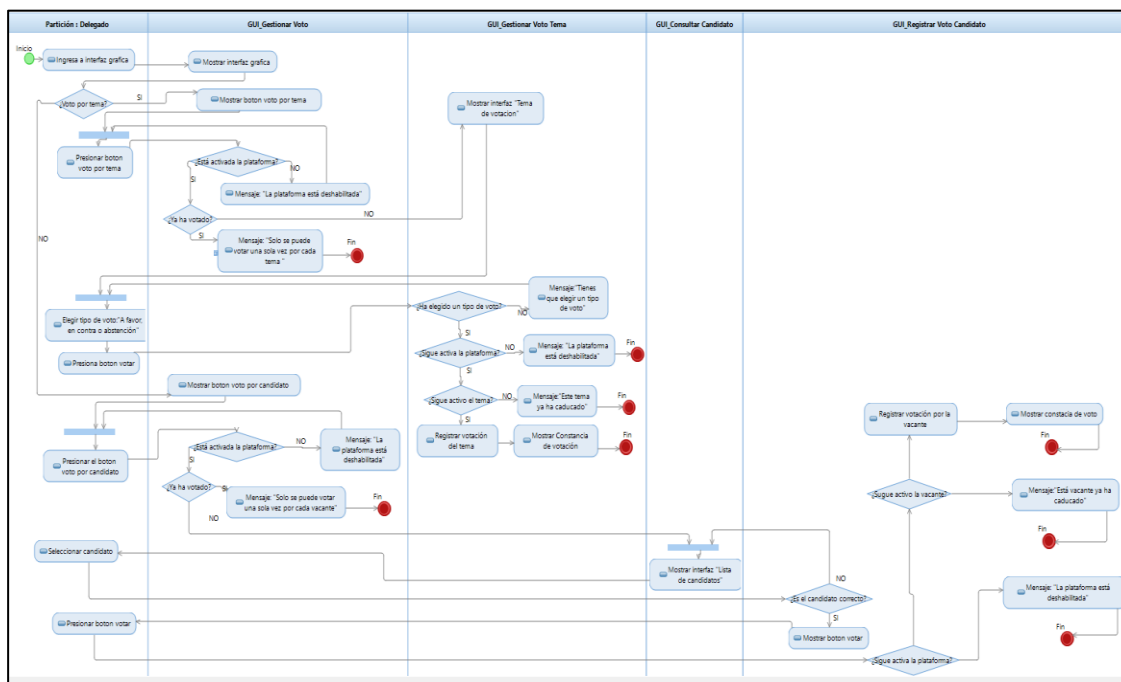
Figura 54
Diagrama de actividades – Gestionar Tema



Fuente: Elaboración propia

Figura 55

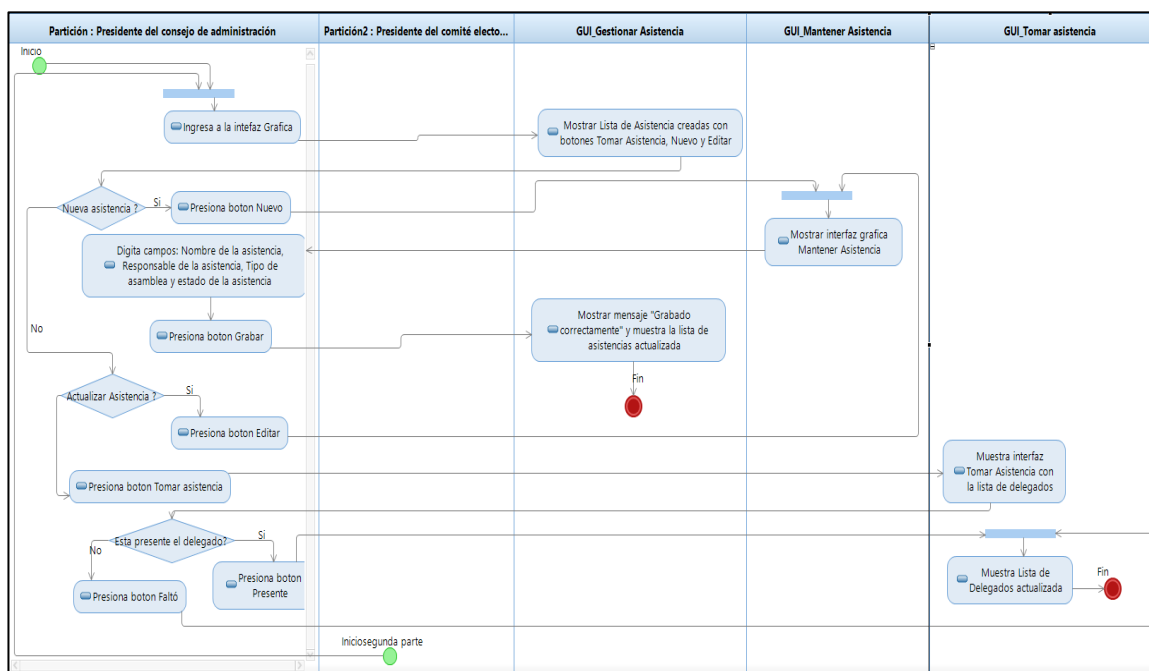
Diagrama de actividades – Gestionar Voto



Fuente: Elaboración propia

Figura 56

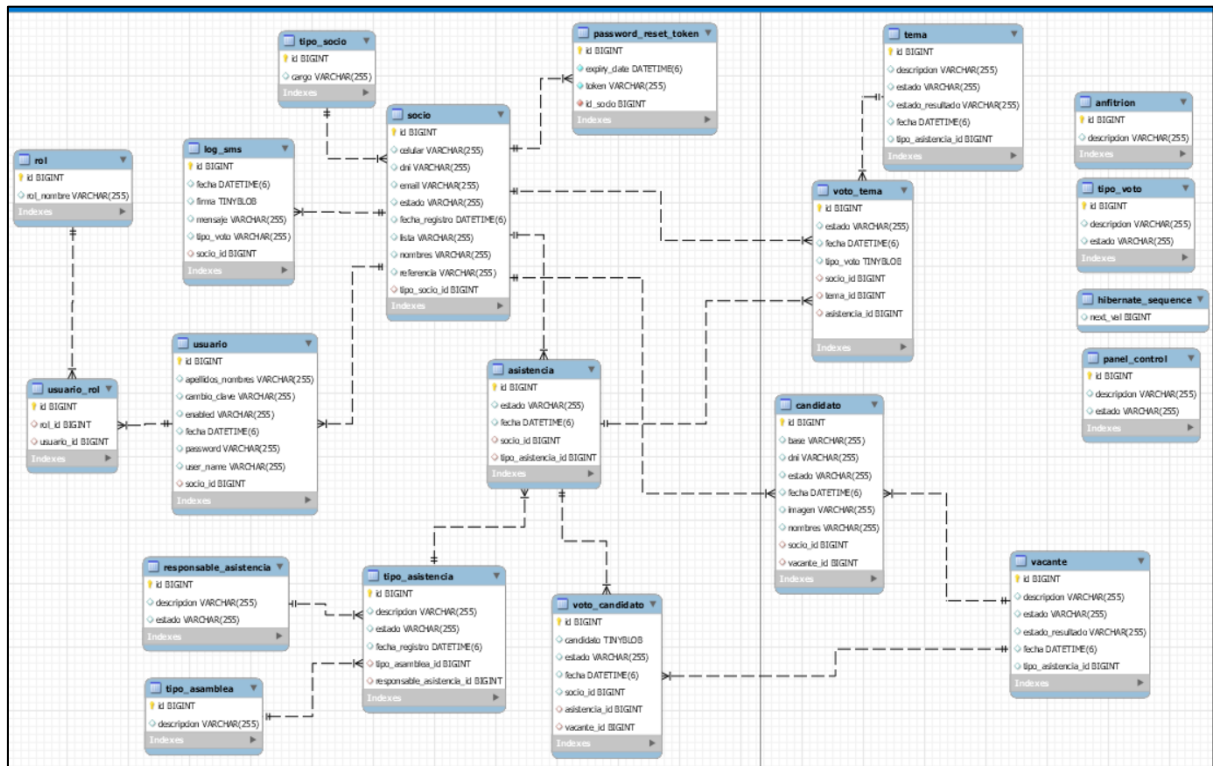
Diagrama de actividades – Gestionar Asistencia



Fuente: Elaboración propia

Figura 57

Esquema de base de datos



Fuente: Elaboración propia

CAPÍTULO IV. DESCRIPCIÓN DE LA EXPERIENCIA

Edgar Ismael Tasayco Altamirano es un bachiller en Ingeniería de Sistemas Computacionales por la Universidad Privada del Norte, con 11 años de experiencia en el rubro de la informática, desarrollando diversos proyectos en tecnología Windows como web, cuento con capacitaciones en diversas entidades reconocidas de Perú como es la Universidad Nacional de Ingeniería (UNI) y la División de Alta Tecnología (DAT) de Cibertec, así como diversos cursos virtuales para el fortalecimiento del conocimiento y actualización de la tecnología.

Vínculo con la empresa Coopsermul 7 de agosto.

Actualmente soy trabajador de la empresa como administrador de sistemas, tengo más de ocho años laborando, donde he desarrollado diversas soluciones informáticas que hoy en día se siguen utilizando para la gestión de los diversos procesos que se realizan, tales como la gestión de préstamos, beneficios por salud y fallecimiento, gestión de planillas, etc. El proyecto de voto electrónico tubo una mayor necesidad en tiempos de pandemia ya que la empresa tiene socios a nivel nacional y se tenían que realizar asambleas virtuales y se necesitaba la herramienta para que los socios puedan votar, la gestión me encarga que desarrolle a modo de servicio una plataforma virtual con todas las medidas de seguridad y transparencia que tenía que tener y estar preparada para cualquier auditoria informática si fuera el caso, y así se realizó, hasta la fecha se han realizado 4 asambleas de delegados (100 delegados por asamblea a nivel nacional)y 1 votación de socios (aprox. 2500 socios). La plataforma ha cumplido con las expectativas a nivel funcional y de seguridad, ha sido un gran acierto establecer mecanismos de seguridad tales como la firma electrónica, así como el cifrado de los votos y constancias además de tener configurado un Crown de seguridad que encarcela cualquier Ip que intente vulnerar la seguridad a nivel de consola

PANTALLAS DEL SISTEMA

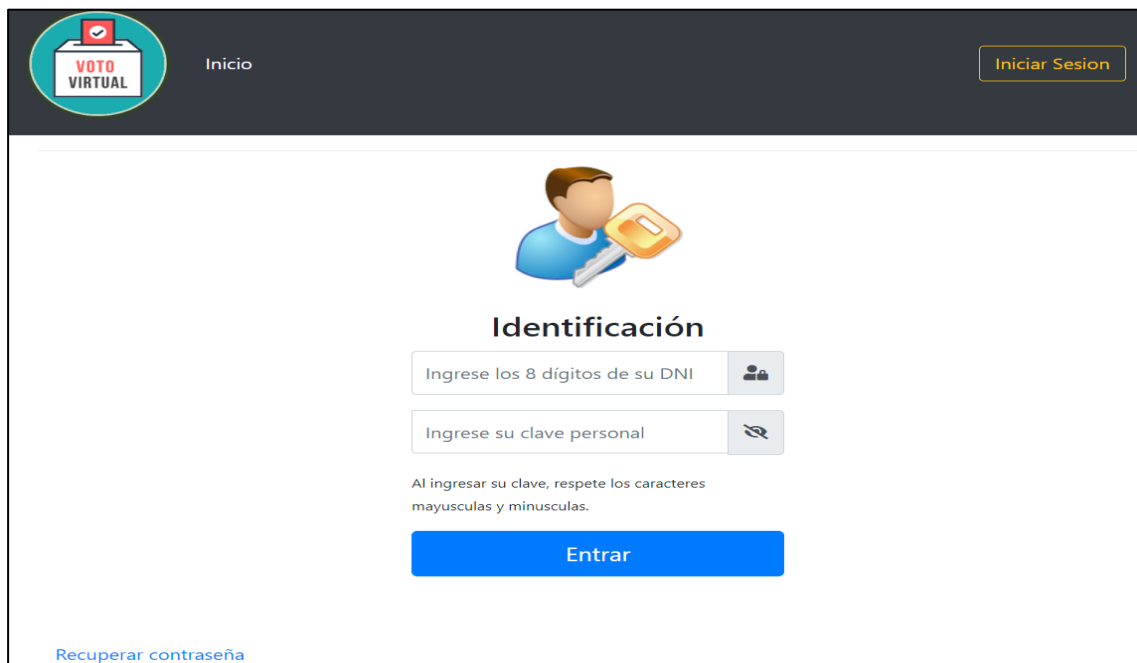
5.1. Presentación

Las pantallas que se muestran a continuación tiene como objetivo mostrar los resultados finales del análisis exhaustivo que se realizó para construir una plataforma de votación que sea intuitiva, transparente y muy segura para el usuario final.

5.2. Pantallas que puede visualizar el administrador de la plataforma

Figura 58

Pantalla Login de usuario



The screenshot shows a web interface for a virtual voting system. At the top left, there is a logo with a ballot box and the text "VOTO VIRTUAL". To its right is the word "Inicio". At the top right, there is a yellow button labeled "Iniciar Sesion". The main content area features a central illustration of a person holding a key. Below this, the heading "Identificación" is displayed. There are two input fields: "Ingrese los 8 dígitos de su DNI" with a person icon, and "Ingrese su clave personal" with a key icon. A note below the fields states: "Al ingresar su clave, respete los caracteres mayusculas y minusculas." A large blue button labeled "Entrar" is positioned below the fields. At the bottom left, there is a blue link that says "Recuperar contraseña".

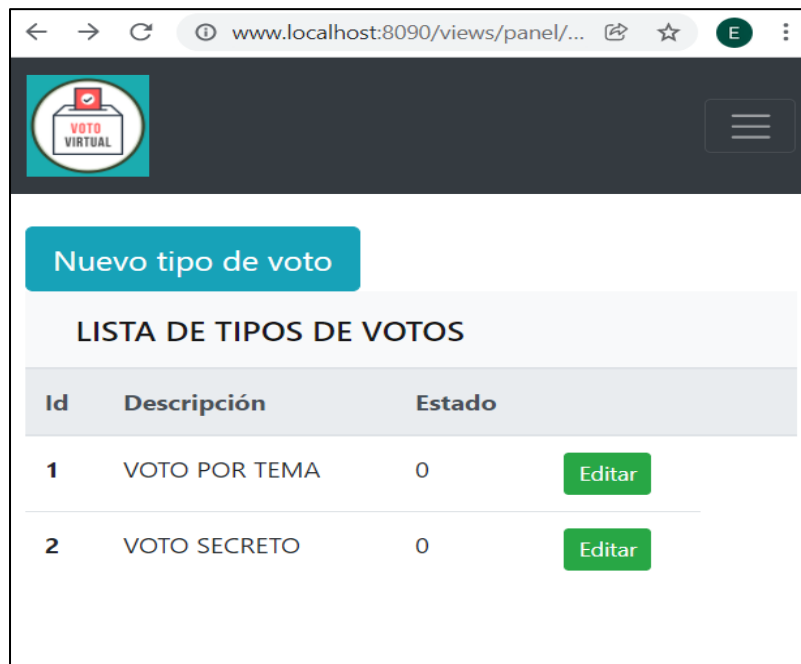
Fuente: Elaboración propia

Figura 59
Pantalla Menú del administrador



Fuente: Elaboración propia

Figura 60
Pantalla Tipos de votaciones



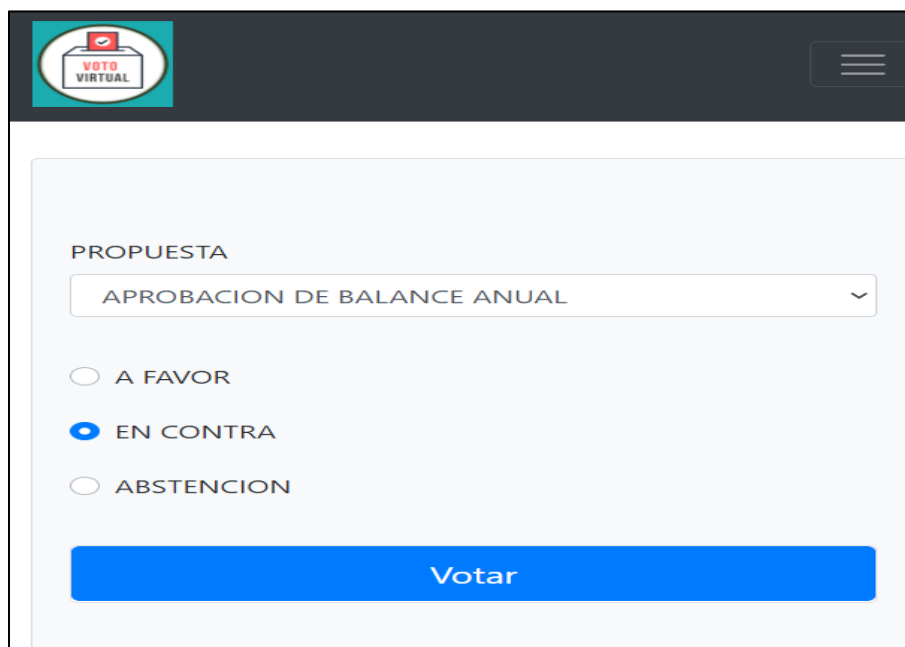
Fuente: Elaboración propia

Figura 61
Pantalla solo voto por Tema



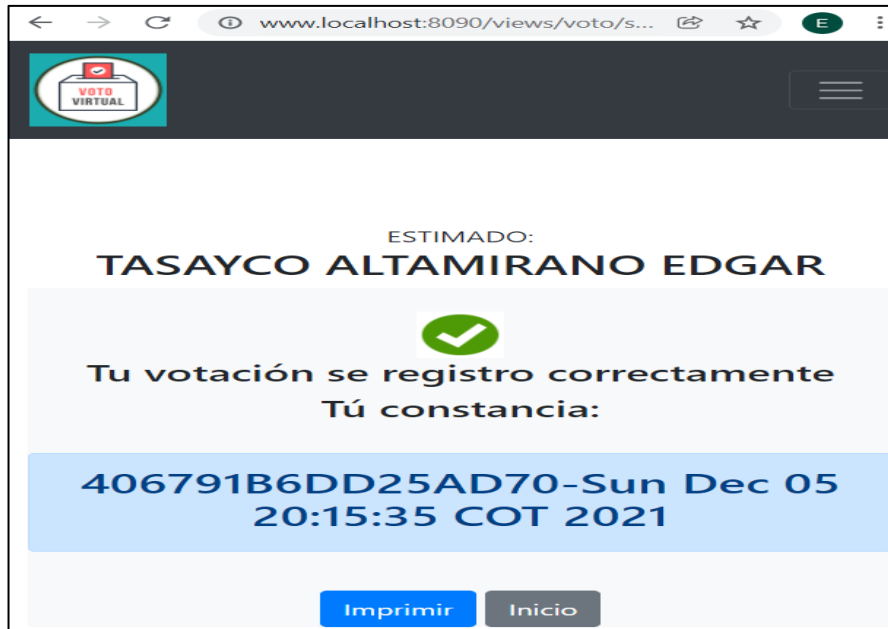
Fuente: Elaboración propia

Figura 62
Pantalla Selección de la Opción a votar por Tema



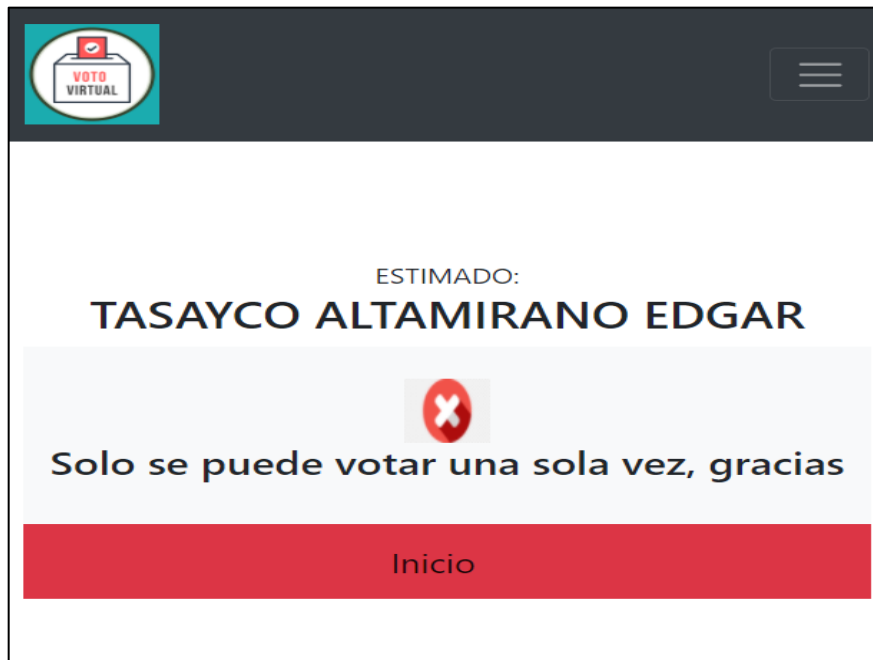
Fuente: Elaboración propia

Figura 63
Pantalla Constancia de voto por Tema



Fuente: Elaboración propia

Figura 64
Pantalla No voto Duplicado



Fuente: Elaboración propia

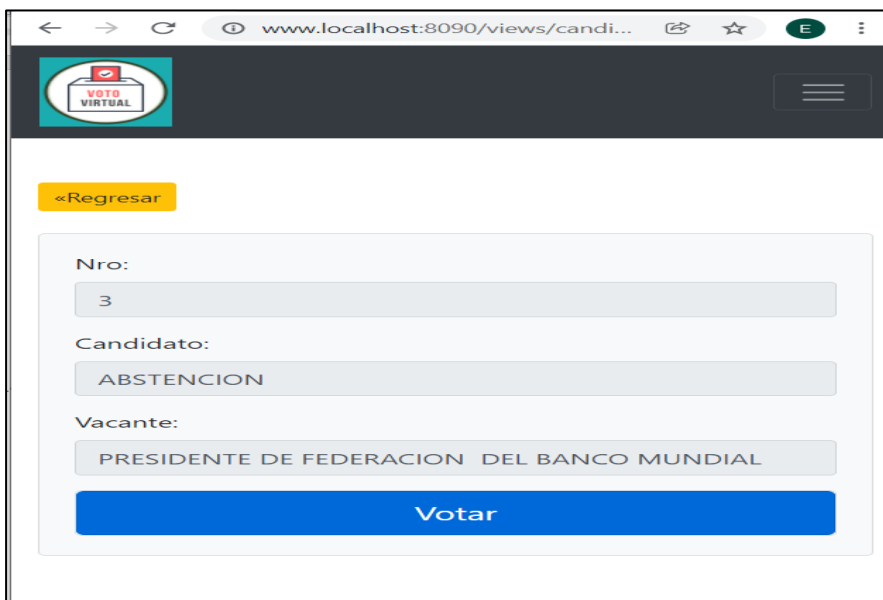
Figura 65
Pantalla Selección de voto por Candidato



NRO.	CANDIDATO	BASE	
1	JUAN PEREZ	LIMA	Seleccionar
2	MAURICIO RIQUELME	LIMA	Seleccionar
3	ABSTENCION	LIMA	Seleccionar

Fuente: Elaboración propia

Figura 66
Pantalla Votar por su Candidato



«Regresar

Nro:
3

Candidato:
ABSTENCION

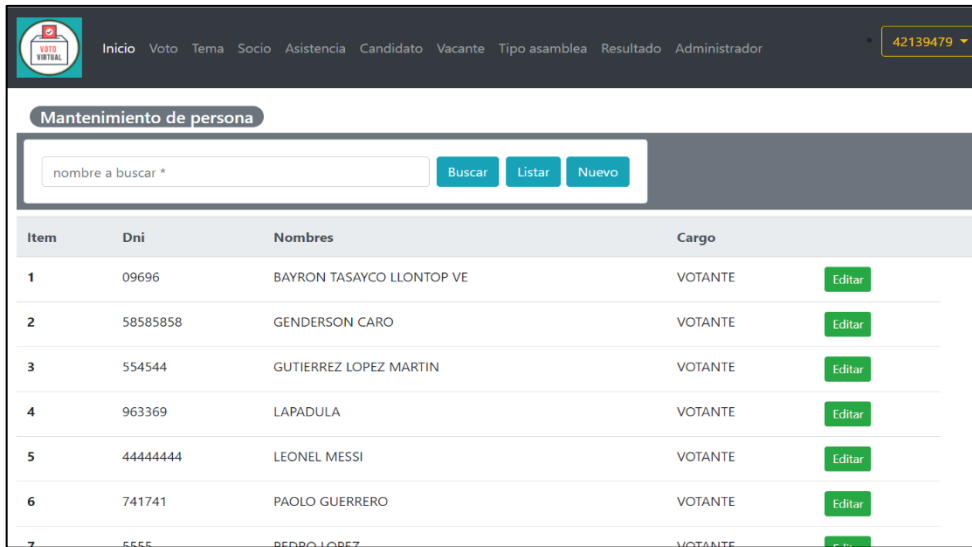
Vacante:
PRESIDENTE DE FEDERACION DEL BANCO MUNDIAL

[Votar](#)

Fuente: Elaboración propia

Nota. Al dar clic en votar, se muestra la constancia de votación igual a la figura 60.

Figura67
Pantalla Gestionar Delegado



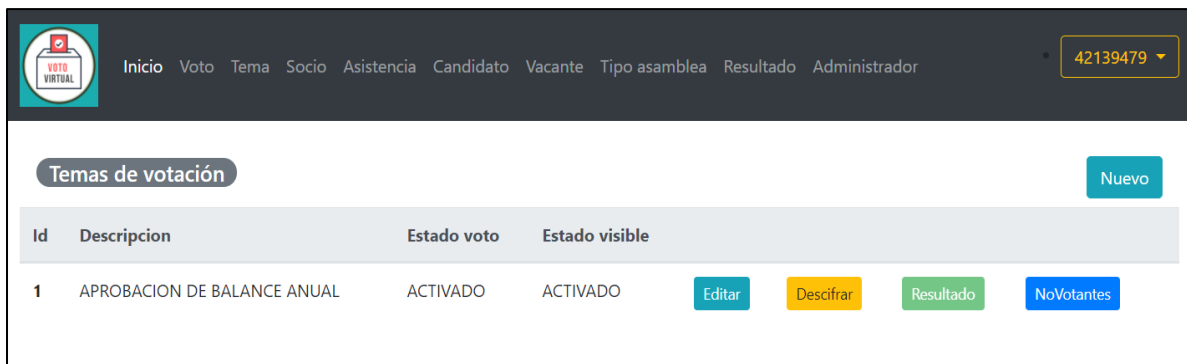
Mantenimiento de persona

nombre a buscar *

Item	Dni	Nombres	Cargo	
1	09696	BAYRON TASAYCO LLONTOP VE	VOTANTE	<input type="button" value="Editar"/>
2	58585858	GENDERSON CARO	VOTANTE	<input type="button" value="Editar"/>
3	554544	GUTIERREZ LOPEZ MARTIN	VOTANTE	<input type="button" value="Editar"/>
4	963369	LAPADULA	VOTANTE	<input type="button" value="Editar"/>
5	44444444	LEONEL MESSI	VOTANTE	<input type="button" value="Editar"/>
6	741741	PAOLO GUERRERO	VOTANTE	<input type="button" value="Editar"/>
7	5555	PEDRO LOPEZ	VOTANTE	<input type="button" value="Editar"/>

Fuente: Elaboración propia

Figura68
Pantalla Gestionar Tema



Temas de votación

Id	Descripcion	Estado voto	Estado visible				
1	APROBACION DE BALANCE ANUAL	ACTIVADO	ACTIVADO	<input type="button" value="Editar"/>	<input type="button" value="Descifrar"/>	<input type="button" value="Resultado"/>	<input type="button" value="NoVotantes"/>

Fuente: Elaboración propia

Figura69
Pantalla Gestionar Asistencia



Mantenimiento de asistencias					Nuevo	
Id	Descripcion	Responsable asistencia	Tipo de asamblea	Estado	Editar	Tomar asistencia
1	VOTACIONES DE PUNTOS DE AGENDA 33334	CONSEJO DE ADMINISTRACION	ASAMBLEA EXTRAORDINARIA	DESACTIVADO	Editar	Tomar asistencia
2	ASISTENCIA PARA ELECCION DE NUEVOS DIRECTIVOS	COMITE ELECTORAL	ASAMBLEA ORDINARIA	ACTIVADO	Editar	Tomar asistencia

Fuente: Elaboración propia

Figura70
Pantalla Gestionar Candidato



Mantenimiento de candidato					Nuevo	
Id	Nro.	Ape. y Nombres	Cargo a postular	Base	Editar	Eliminar
1	1	JUAN PEREZ	PRESIDENTE DE FEDERACION DEL BANCO MUNDIAL	LIMA	Editar	Eliminar
2	2	MAURICIO RIQUELME	PRESIDENTE DE FEDERACION DEL BANCO MUNDIAL	LIMA	Editar	Eliminar
3	3	ABSTENCION	PRESIDENTE DE FEDERACION DEL BANCO MUNDIAL	LIMA	Editar	Eliminar
4	777	CARVALO CONTAGIANI	PRESIDENCIA DEL PERU		Editar	Eliminar
5	888	GENDERSON JUY	DT DEL PERU 2022		Editar	Eliminar
6	7878	PS GUARDIOLA	DT DEL PERU 2022		Editar	Eliminar

Fuente: Elaboración propia

Figura71
Pantalla Gestionar Vacante



Id	Descripcion	Estado voto	Resultado visible					
1	PRESIDENTE DE FEDERACION DEL BANCO MUNDIAL	ACTIVADO	DESACTIVADO	Editar	Descifrar	Resultado	NoVotantes	Votantes
2	PRESIDENCIA DEL PERU	DESACTIVADO	DESACTIVADO	Editar	Descifrar	Resultado	NoVotantes	Votantes
3	PRESIDENTE DE BARRANQUITO	DESACTIVADO	DESACTIVADO	Editar	Descifrar	Resultado	NoVotantes	Votantes
4	DT DEL PERU 2022	DESACTIVADO	DESACTIVADO	Editar	Descifrar	Resultado	NoVotantes	Votantes

Fuente: Elaboración propia

Figura72
Pantalla Resultado de Tema



Nro	Descripción	Votos		
1	A FAVOR	0	Descifrar	Mostrar
2	EN CONTRA	1	Descifrar	Mostrar
3	ABSTENCION	0	Descifrar	Mostrar
4	PRUEBA	0	Descifrar	Mostrar
0	NO VOTANTE	4	Descifrar	Mostrar
TOTAL:		5		


Fuente: Elaboración propia

Figura73
Pantalla Resultado de Candidato - Voto Secreto

RESULTADO DE VOTACIÓN DE LA VACANTE: PRESIDENTE DE FEDERACION DEL BANCO MUNDIAL		
Nro.	Candidato	Votos
1	JUAN PEREZ	0
2	MAURICIO RIQUELME	0
3	ABSTENCION	1
0	NO VOTANTE	4
		TOTAL: 5

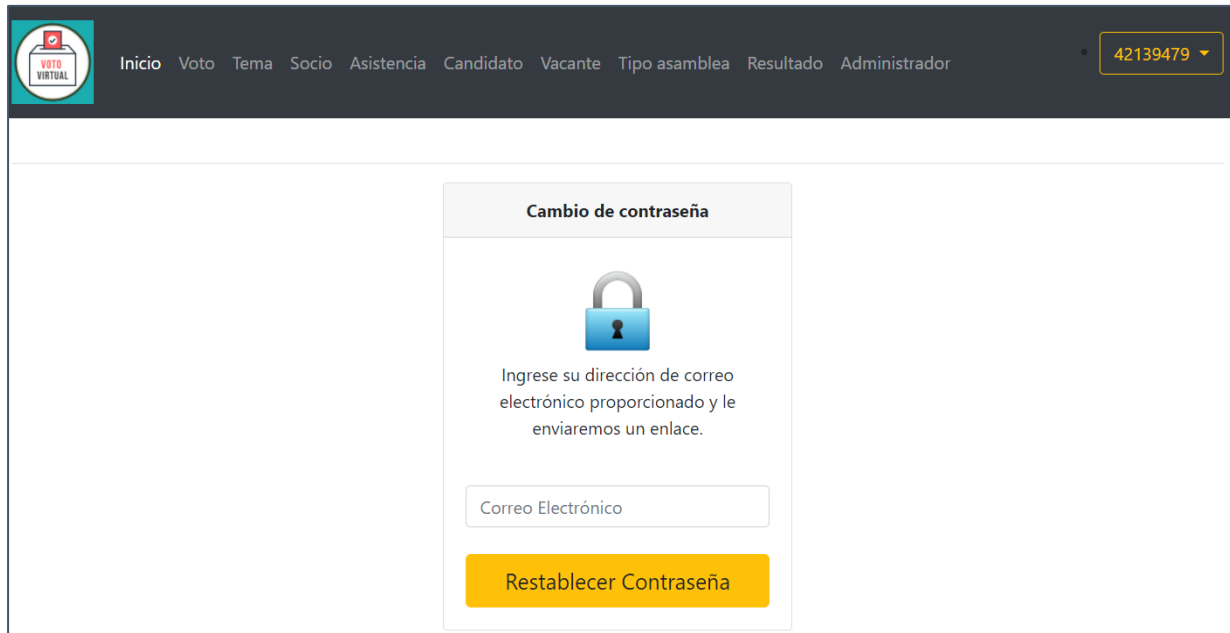
Fuente: Elaboración propia

Figura74
Pantalla de Firma Digital - Criptografía asimétrica

 Inicio Voto Tema Socio Asistencia Candidato Vacante Tipo asamblea Resultado Administrador 42139479 ▾			
Listado del log de votaciones:			
Id	Votante	Firma	Fecha
1	TASAYCO ALTAMIRANO EDGAR	13857FFA43B92A23	2021-06-11 22:56:50.373
2	TASAYCO ALTAMIRANO EDGAR	406791B6DD25AD70	2021-06-11 23:15:46.971
3	PEDRO LOPEZ	13BECDB00B3C59F8	2021-06-11 23:21:36.934
4	PEDRO LOPEZ	6BBB9BD5E3573144	2021-06-15 01:57:13.148
5	TASAYCO ALTAMIRANO EDGAR	13857FFA43B92A23	2021-06-15 02:04:55.748
6	PEDRO LOPEZ	6CCD60B9493DB1F2	2021-06-15 02:06:08.481
7	TASAYCO ALTAMIRANO EDGAR	8977E80F31CEE20	2021-06-15 02:06:40.414
8	GENDERSON CARO	6CCD60B9493DB1F2	2021-06-15 02:07:21.887

Fuente: Elaboración propia

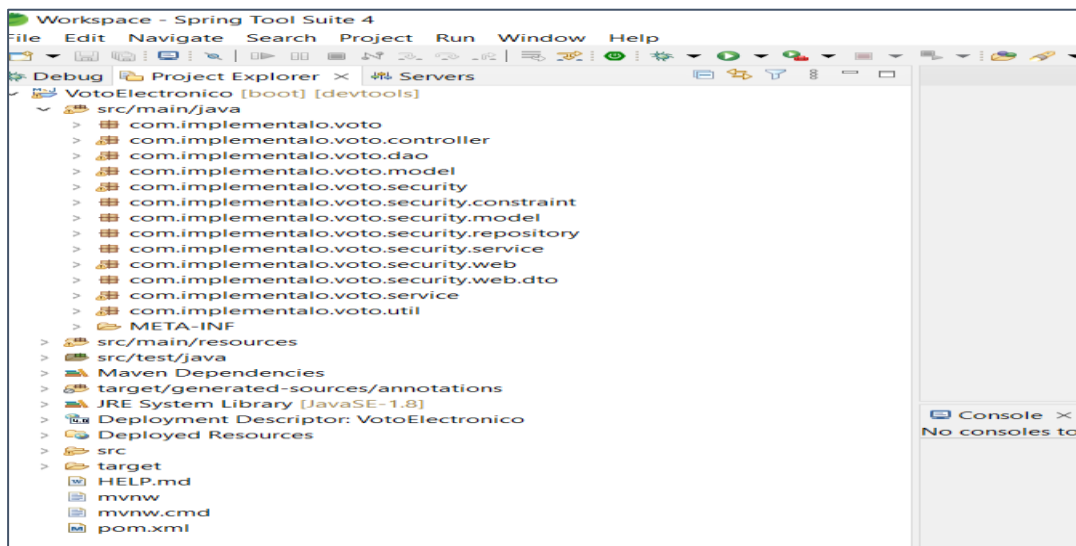
Figura75
Pantalla Cambio de Contraseña



Fuente: Elaboración propia

FUENTES DEL SISTEMA

Figura 76
Fuente Árbol de paquetes



Fuente: Elaboración propia

Figura 77
Fuente Pom Xml - Referencia maven

```
*VotoElectronico/pom.xml x
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"
5   <modelVersion>4.0.0</modelVersion>
6   <parent>
7     <groupId>org.springframework.boot</groupId>
8     <artifactId>spring-boot-starter-parent</artifactId>
9     <version>2.2.6.RELEASE</version>
10    <relativePath /> <!-- lookup parent from repository -->
11  </parent>
12  <groupId>com.implementalo</groupId>
13  <artifactId>VotoElectronico</artifactId>
14  <version>0.0.1-SNAPSHOT</version>
15  <packaging>war</packaging>
16  <name>VotoElectronico</name>
17  <description>Voto electronico para asambleas</description>
18
19  <properties>
20    <java.version>1.8</java.version>
21  </properties>
22
23  <dependencies>
24    <dependency>
25      <groupId>org.springframework.boot</groupId>
26      <artifactId>spring-boot-starter-data-jpa</artifactId>
27    </dependency>
28    <dependency>
29      <groupId>org.springframework.boot</groupId>
30      <artifactId>spring-boot-starter-thymeleaf</artifactId>
31    </dependency>
32    <dependency>
33      <groupId>org.springframework.boot</groupId>
34      <artifactId>spring-boot-starter-web</artifactId>
35    </dependency>
36
37    <dependency>
38      <groupId>org.springframework.boot</groupId>
39      <artifactId>spring-boot-devtools</artifactId>
40    <scope>runtime</scope>
```

Fuente: Elaboración propia

Figura 78*Fuente Aplicando Hibernate*

```
@Entity
@Table(name="asistencia")
public class Asistencia implements Serializable{

    @Id
    @Column(name="id")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long id;

    @Temporal(TemporalType.TIMESTAMP)
    @Column(name="fecha")
    private Date fecha;

    @Column(name="estado" ,length=2)
    private String estado="1";

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name="socio_id")
    private Socio socio ;

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name="tipo_asistencia_id")
    private TipoAsistencia tipoAsistencia;

    @OneToMany(mappedBy = "asistencia")
    private Set<VotoCandidato>votoCandidato;

    @OneToMany(mappedBy = "asistencia")
    private Set<VotoTema>votoTema;

    public Long getId() {
        return id;
    }
}
```

Fuente: Elaboración propia

Figura 79*Fuente Aplicando Spring Data*

```
4
5* import java.util.List;
6
7 @Repository
8 public interface SocioDao extends CrudRepository<Socio, Long> {
9
10
11     @Query("select s from Socio s where s.lista=1 and estado=1")
12     List<Socio> findBySocioVotante();
13
14     Socio findByEmail(String email);
15
16     Socio findByDni(String dni);
17
18     @Query("select s from Socio s where s.nombres like %?1% ")
19     public List<Socio>listarSocio(String nombres);
20
21
22 }
```

Fuente: Elaboración propia

Figura 80

Fuente Aplicando Seguridad a nivel de Url

```
@Override
protected void configure(HttpSecurity http) throws Exception {

    http.authorizeRequests()
        .antMatchers("/index",
            "/home",
            "/",
            "/css/**",
            "/bcrypt/**",
            "/js/**",
            "/images/**",
            "/webfonts/**",
            "/sass/**",
            "/assets/**",
            "/forgot-password**", // 6
            "/reset-password**").permitAll() // 7

        .anyRequest().authenticated()
        .and()
        .formLogin()
        .successHandler(successMessage)
        .loginPage("/login")
        .permitAll()
        .and()
        .logout()
        .invalidateHttpSession(true) // 1
        .clearAuthentication(true) // 2
        .logoutRequestMatcher(new AntPathRequestMatcher("/logout")) // 3
        .logoutSuccessUrl("/login?logout") // 4
        .permitAll();

    http.sessionManagement()
        .maximumSessions(1) // 3
        .maxSessionsPreventsLogin(true) // 4
        .expiredUrl("/logout") // 5
        .sessionRegistry(sessionRegistry());
}
}
```

Fuente: Elaboración propia

Figura 81

Fuente Seteando Clave privada - corografía asimétrica

```
public void generaPrimos()
{

    //ELIGIENDO LOS DOS NROS PRIMOS
    p=new BigInteger("101303735110271681"); //Esto se podria tener como variable
    q=new BigInteger("1555073079"); //Esto se podria tener como variable

}

public void generaClaves()
{
    // n = p * q HALLANDO EL VALOR DE N:
    n = p.multiply(q);
    // totient = (p-1)*(q-1)
    totient = p.subtract(BigInteger.valueOf(1)); //ACA LE RESTA 1 A P
    totient = totient.multiply(q.subtract(BigInteger.valueOf(1)));

    e=new BigInteger("65537");
    //d = e.modInverse(totient);
    d=new BigInteger("147321598812486");
}
}
```

Fuente: Elaboración propia

Figura82

Fuente Seguridad a nivel de roles

```
@Secured({"ROLE_USER", "ROLE_ADMIN"})
@RequestMapping("/listarElecto/{id}")
public String listarPorCargo(@PathVariable("id") Long id, Model model) {

    Vacante cargoSeleccionado= _cargoService.findByCargoId(id);

    List<VotoCandidato>votosPorVacante= _eleccionService.votantesPorVacante(id);
    //List<TipoVoto>tipoVoto= _tipoVotoService.listarTipoVoto();
    List<Candidato> candidato = _candidatoService.findByVacante(id);
    //List<Socio> cantidadSocioVotante= _socioVotante.findBySocioVotante();
    List<Asistencia> cantidadSocioVotante= _asistenciaService.buscarTipoAsistenciaAndEstado(cargoSeleccionado.getTipoAsistenciaId(),
    //Asistencia c= _asistenciaService.buscarSocioPorAsistencia(null, null, null)

    //Validar que no exista cargo activo antes de descubrir los votos

    Vacante cargo= _cargoService.findByCargoActiva();
    if (cargo!=null) {
        return "redirect:/views/ok/invalido";
    }

    //TipoAsistencia asamblea= _tipoAsistenciaService.tipoAsistenciaActiva();

    int nroVotos=0,nroVotantes =0,totalNoVotante=0 ;
    double porcentaje=0.00,porcentajeNovoto=0.0;
    nroVotantes=cantidadSocioVotante.size();
    nroVotos=votosPorVacante.size();
    totalNoVotante=nroVotantes - nroVotos;
```

Fuente: Elaboración propia

Figura83

Fuente Plantilla de menú principal - Template thymeleaf

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
      xmlns:sec="http://www.thymeleaf.org/extras/spring-security">
<head th:fragment="head">
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<title>Voto virtual</title>
<link rel="stylesheet" th:href="@{/css/bootstrap.min.css}">
<link rel="stylesheet" th:href="@{/css/signin.css}">
<link rel="stylesheet" th:href="@{https://use.fontawesome.com/releases/v5.8.1/css/all.css}"/>
</head>
<body>
<!-- BARRA DE NAVEGACION -->
<header th:fragment="header">

    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container">
    <a class="navbar-brand" th:href="@{/}"> </a>
    <button class="navbar-toggler" type="button" data-toggle="collapse"
            data-target="#navbarSupportedContent"
            aria-controls="navbarSupportedContent" aria-expanded="false"
            aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
    </button>
    <!-- ENLACES DEL MENU -->
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
    <li class="nav-item active"><a class="nav-link"
            th:href="@{/}">Inicio <span class="sr-only">(current)</span>
    </a></li>
    <li class="nav-item" sec:authorize="isAuthenticated()"><a
            class="nav-link" th:href="@{/views/votar/miVoto}">Voto</a></li>

    <li class="nav-item" sec:authorize="hasRole('ROLE_ADMIN')"><a
            class="nav-link" th:href="@{/views/propuesta/Listar}">Tema</a></li>

    <li class="nav-item" sec:authorize="hasRole('ROLE_ADMIN')"><a
            class="nav-link" th:href="@{/views/socio/Listar}">Socio</a></li>
    <li class="nav-item" sec:authorize="hasRole('ROLE_ADMIN')"><a
            class="nav-link" th:href="@{/views/tipoAsistencia/Listar}">Asistencia</a></li>
    </ul>
    </div>
    </div>
    </nav>

```

Fuente: Elaboración propia

RESULTADOS

El presente trabajo desarrollado se obtuvieron los siguientes resultados:

- Con la utilización de una plataforma de votación se podrá realizar votaciones de manera rápida, ello concuerda con su proyecto de investigación de Aradiel Castañeda Hilario.
- La utilización de la criptografía asimétrica tiene una función esencial en la plataforma de votación, pues esto nos garantiza el no repudio que debe existir en este tipo de comicios electorales, poder develar resultados sin saber quién o quienes votaron, permite respetar la opción del voto que el elector eligió, ello concuerda con Moroco (2010).
- Creando una interfaz simple e intuitiva donde el elector pueda reconocer que hacer para poder realizar su votación es fundamental para evitar posible reclamos o quejas de interfaces nada amigables, ello concuerda con Schmidt-Peralta y Gutiérrez-Alfaro (2015).
- Desarrollar una aplicación utilizando algún patrón de arquitectura que nos permita separar la lógica del negocio es fundamental en este tipo de procesos, existe mucha seguridad que se maneja en la parte del backend, así como la seguridad a nivel de métodos, esto tiene gran concordancia con Vaca et al. (2019).
- La plataforma ha desarrollado 3 votaciones en asambleas virtuales donde participan los 100 delegados a nivel nacional y 1 votación a nivel nacional con aproximadamente 2500 socios, esto se puede realizar en cumplimiento de las leyes de salubridad que brinda el estado peruano, es por ello que las leyes dentro de un territorio nacional otorgan respaldo a ejercer cualquier actividad legal, esto tiene gran concordancia con Monserrate (2021).

CONCLUSIONES

de haber planificado, desarrollado y puesta en funcionamiento podemos concluir, que para un correcto funcionamiento de la plataforma de votación tanto a nivel de seguridad, funcionalidad y transparencia, se tiene que haber ejercido un correcto análisis a nivel de arquitectura y desarrollo, así como también de aquellos aspectos externos que influyen de manera directa en este tipo de procesos, tales como el respeto por las leyes a las que se rigen este tipo de servicio.

La utilización de una adecuada tecnología que te brinde diversas soluciones simples y segura, ayudará de manera eficiente para acortar el tiempo de desarrollo, en este proyecto la utilización de framework de Spring fue muy eficiente al brindar diversos módulos para diferentes etapas del proyecto.

Una correcta planificación basado en la seguridad que otorgue tranquilidad al elector debería ser un objetivo principal, debemos asegurar nuestra aplicación a nivel de los protocolos Http y también a nivel de ataques de consola.

La metodología RUP nos permite mapear las fases del proyecto, para tener un mejor panorama del comportamiento que tendrá nuestra aplicación pues en aquí podemos establecer lo

BIBLIOGRAFÍA

- Apache Tomcat. (s.f.). *Apache Tomcat*. Obtenido de Apache Tomcat: <http://tomcat.apache.org/>
- Barrientos, M. (2021). *Mecanismo de bloqueo a conexiones remotas que intentan accesos por fuerza bruta para Nova-LTSP*.
- Fernando Sánchez, T., & Wong Portillo, L. (2010). Mejorando las debilidades de RUP. *Revista de Investigación de Sistemas e Informática*, 50.
- García Hernández, S. D. (2006). *Flujo de trabajo del análisis y diseño del Rup*.
- Hibernate. (s.f.). *HQL: El lenguaje de consulta de Hibernate*. Obtenido de <https://docs.jboss.org/hibernate/core/3.6/reference/es-ES/html/queryhql.html>
- Hibernate. (24 de 05 de 2021). *Guía de inicio de Hibernate*. Obtenido de Prefacio: https://docs.jboss.org/hibernate/orm/5.4/quickstart/html_single/
- Let's Encrypt. (s.f.). *Documentación*. Obtenido de <https://letsencrypt.org/es/docs/>
- Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de software*.
- Marini, E. (2012). *El Modelo Cliente/Servidor*.
- Martínez, A., & Martínez, R. (2014). *Guía a Rational Unified Process*.
- Maven. (15 de 01 de 2021). *¿Que es Maven?* Obtenido de <https://maven.apache.org/what-is-maven.html>
- Méndez Gonzáles, G. (2016). *Aprende a Desarrollar con Spring Framework: 2ª Edición*. IT Campus Academy.
- Nevado Cabello, V. (2010). *Introducción a las bases de datos relacionales*. Madrid: Vision Libros.
- Onpe. (3 de 11 de 2021). *Innovaciones Tecnológicas*. Obtenido de Minisite de Voto Electrónico: <https://www.onpe.gob.pe/elecciones/innovaciones-tecnologicas/>

Onpe. (3 de 11 de 2021). *Voto electronico*. Obtenido de Practique aquí:

<https://www.venp.pe/venp/inicio.do>

Panizo, A. L. (2010). *Aspectos Tecnológicos del voto electrónico*. España: Fundación CEPS.

Obtenido de <https://buleria.unileon.es/handle/10612/3144>

Pesado, P. M. (2015). VOTO ELECTRÓNICO: un desafío de innovación y eficiencia. *Bit & Byte*.

Spring. (09 de 12 de 2020). *Tecnologías centrales*, 5.3.2. Obtenido de 1.10.1. @Componenty más anotaciones de estereotipos: <https://docs.spring.io/spring->

[framework/docs/current/reference/html/core.html#beans-annotation-config](https://docs.spring.io/spring-framework/docs/current/reference/html/core.html#beans-annotation-config)

Spring. (2016). *Documentación de referencia de Spring Framework*. Obtenido de Inyección de dependencia e inversión de control: <https://docs.spring.io/spring->

[boot/docs/current/reference/htmlsingle/#dependency-versions-properties](https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#dependency-versions-properties)

Spring. (s.f.). *Referencia de spring security*. Obtenido de Protección contra ataques CSRF:

<https://docs.spring.io/spring-security/site/docs/5.4.2/reference/html5/#exploits>

Spring security. (s.f.). *Referencia de Spring Security*. Obtenido de Autenticación:

<https://docs.spring.io/spring-security/site/docs/5.4.2/reference/html5/#authentication>

Spring. (s.f.). *Spring Data JPA 2.4.3*. Obtenido de <https://spring.io/projects/spring-data-jpa>

Spring. (s.f.). *Spring Tools 4*. Obtenido de Proyectos de Spring: <https://spring.io/tools>

ZeroSSL. (s.f.). *Certificados SSL*. Obtenido de <https://zerossl.com/features/certificates/>

ANEXOS

Anexo 1

Publicación de convocatoria

Circulación Nacional **Avisos**

AVISOS LEGALES **JUDICIALES**



CONVOCATORIA A ASAMBLEA GENERAL ORDINARIA DE DELEGADOS

El Consejo de Administración de la Cooperativa de Servicios Múltiples "7 de Agosto Ltda.", por acuerdo de sesión de dicho Consejo, del 06 de Noviembre del 2020, de conformidad al artículo 30 numeral 16 del Texto Único Ordenado de la Ley General de Cooperativas (D.S. N° 074-90-TR), concordante con el Artículo 57, literal "k" del Estatuto vigente y lo establecido por el Decreto de Urgencia N°075-2020, convoca a los Delegados a la **ASAMBLEA GENERAL ORDINARIA**, no presencial - virtual, que se llevara a cabo el día **Domingo 29 de Noviembre del 2020**, a través de la plataforma MEET, enlace/ <https://meet.google.com/nrf-piyn-nii>, ubicado en la Calle Presa N° 133. Distrito del Rimac, Provincia y Departamento de Lima:

Primera Citación : 09.00 horas
Segunda Citación : 10.00 horas

AGENDA:

- 1.- Memoria Anual del Consejo de Administración 2019. ✓
- 2.- Memoria Anual del Consejo de Vigilancia 2019. ✓
- 3.- Memoria Anual del Comité Electoral Nacional 2019. ✓
- 4.- Informe Económico Financiero Periodo 2019. ✓
- 5.- Aprobación del Balance y Estados Financieros 2019. ✓
- 6.- Aprobación del Presupuesto 2020. ✓
- 7.- Ratificación del Aporte Mínimo Mensual. ✓
- 8.- Aprobación de Dietas y Gastos de Representación en aplicación al Art.16 y 45 letra d) del Estatuto Vigente. ✓
- 9.- Renovación de Tercios Dirigencales de los Consejos y Comités. ✓

Rimac, 12 de Noviembre del 2020.

<p>RAFAEL CHUQUIVILCA CHACON DNI N° 08709985 PRESIDENTE CONSEJO DE ADMINISTRACION</p>	<p>SEGUNDO R. BUSTAMANTE GOMEZ DNI N°09480570 SECRETARIO CONSEJO DE ADMINISTRACION</p>
--	---

AV

A los

Se c

Accid

UNIC

"Ley

novie

ta me

zoom

UrQil

ID de

Códi

de ac

celeb

asam

medi

de sc

1) A

de

2) A

3) D

pr

y

De n

segu

del 2

para

refer

los s

págit

com

Asim

lo cu

corre

com

elec

Adic

de l

recc

3er.

Com

Los

pers

el d

Anexo 2

Ley 31029 - para voto electrónico en Cooperativas

El Peruano

FUNDADO EL 22 DE OCTUBRE DE 1825 POR EL LIBERTADOR SIMÓN BOLÍVAR

El Perú, mediante el uso de medios tecnológicos y telemáticos y de comunicaciones, dará curso los registros electrónicos de dichas organizaciones cooperativas para reconozcan la posibilidad de convocar y celebrar sesiones presenciales.

La convocatoria, frecuencias de sesiones y la adopción de acuerdos se rigen por lo señalado en el estatuto de la cooperativa para sesiones presenciales; siempre y cuando dichos requisitos sean aplicables para la realización de una sesión no presencial. Para el caso de la asamblea general, el quorum mínimo para su instalación será del cincuenta por ciento más uno de sus miembros.

Los acuerdos, que se adopten en el marco de la presente disposición complementaria transitoria, tendrán plena validez, eficacia y los mismos efectos que los acuerdos adoptados en sesiones de carácter presencial.

Vencido el plazo establecido en el primer párrafo, las organizaciones cooperativas creadas o por crearse deberán contemplar de forma expresa la facultad de realizar sesiones no presenciales y los demás requisitos que se requieran.

SEGUNDA. Reglamento de Inscripción de Cooperativas

La Superintendencia Nacional de los Registros Públicos, en un plazo de 120 días calendario contados desde la publicación de la presente ley, aprueba el Reglamento de Inscripción de Cooperativas, para regular la inscripción de actos, considerando los pronunciamientos emitidos a la fecha y que se encuentren concordantes con la presente ley, la Ley General de Cooperativas y la Ley General de Sociedades, y demás normas que resulten aplicables para las cooperativas.

En un plazo no menor de treinta (30) días antes de la fecha prevista para su entrada en vigencia, debe publicarse el proyecto del Reglamento de Inscripción de Cooperativas.

En tanto y en cuanto la Superintendencia Nacional de los Registros Públicos no apruebe el reglamento de inscripción de cooperativas, es de aplicación el Reglamento de Inscripciones del Registro de Personas Jurídicas aprobado mediante Resolución 038-2013-SUNARP-SN, del 19 de febrero de 2013. Comuníquese al señor Presidente de la República para su promulgación.

En Lima, a los veinticuatro días del mes de junio de dos mil veinte.

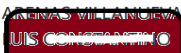







MANUEL MERINO DE LAMA
Presidente del Congreso de la República

LUIS ALBERTO VALDEZ FARIÁS
Primer Vicepresidente del Congreso de la República

AL SEÑOR PRESIDENTE DE LA REPÚBLICA

Anexo 3

Rastros de firma digital

Detalle de log Sms				
Id	Estado	Votante	Firma	Fecha
1	OK dest:5192727357		2D19824AE3247070-6CCD60B9493DB1F2-406791B6DD25AD70	2021-02-27 08:03:02.902
2	OK dest:51947711008		2D19824AE3247070-13BECDB00B3C59F8-67B6B2732F701DA2-8B91315A580D0FEB	2021-02-27 08:04:18.209
3	OK dest:5190571186		2D19824AE3247070-6BBB9BD5E3573144-2D19824AE3247070-6BBB9BD5E3573144	2021-02-27 08:04:52.396
4	OK dest:5198111087		6BBB9BD5E3573144-67B6B2732F701DA2-13BECDB00B3C59F8	2021-02-27 08:05:37.008
5	OK dest:51991119221		2D19824AE3247070-406791B6DD25AD70-6BBB9BD5E3573144-67B6B2732F701DA2	2021-02-27 08:06:01.609
6	OK dest:5111111366		406791B6DD25AD70-406791B6DD25AD70-487E7BFF98A51C62	2021-02-27 08:06:54.26
7	OK dest:5111111236		2D19824AE3247070-6BBB9BD5E3573144-13857FFA43B92A23	2021-02-27 08:06:59.365
8	OK dest:51927270209		2D19824AE3247070-8B91315A580D0FEB-406791B6DD25AD70-406791B6DD25AD70	2021-02-27 08:07:17.056

Anexo 4

Resultado de votación - para el Comité electoral

[«Regresar](#)

RESULTADO DE VOTACIÓN :
COMITE ELECTORAL 29-11-2020

Nro.	Candidato	Votos
1	JORGE LUIS REYES TARAZONA	63
2	OSCAR VELA VELA	20
3	ABSTENCION	1
0	NO VOTANTE	0
TOTAL:		84

Anexo 5

Resultado de votación - para el Comité de vigilancia

[«Regresar](#)

RESULTADO DE VOTACIÓN :
CONSEJO DE VIGILANCIA 29-11-2020

Nro.	Candidato	Votos
1	PUZA TITO PRISCIALIANO	29
2	BADAJEZ BUSTAMANTE FIDENCIANO	55
3	ABSTENCION	0
0	NO VOTANTE	0
TOTAL:		84

Anexo 6

Resultado de votación - para el Consejo de administración

«Regresar»

RESULTADO DE VOTACIÓN :
CONSEJO DE ADMINISTRACION 29-11-2020

Nro.	Candidato	Votos
1	SEGUNDO BEINGOLEA VEGA	22
2	JOHNNY PALOMINO GONZALES	31
3	WALTER VILLANUEVA MAZA	31
4	ABSTENCION	0
0	NO VOTANTE	0
TOTAL:		84

Anexo 7

Resultado voto por tema - Oportunidad de voto

«Regresar»

RESULTADO DE VOTACIÓN :
**QUE LA VOTACION SECRETA SE REALICE EN BLOQUE PARA LA
RENOVACION DE TERCIOS DIRIGENCIALES EN UN TIEMPO DE 10 MINT. Y
UNA 2DA OPORTUNIDAD DE 5 MINT. A LOS QUE NO VOTARON.**

Nro	Descripción	Votos	Descifrar	Descubrir
1	A FAVOR	80	Descifrar	Descubrir
2	EN CONTRA	1	Descifrar	Descubrir
3	ABSTENCION	2	Descifrar	Descubrir
0	NO VOTANTE	1	Descifrar	Descubrir
TOTAL:		84		

Anexo 8

Resultado de votación - aprobación de dietas

[«Regresar](#)

RESULTADO DE VOTACIÓN :
APROBACION DE DIETAS Y GASTOS DE REPRESENTACION.

Nro	Descripción	Votos		
1	A FAVOR	61	Descifrar	Descubrir
2	EN CONTRA	3	Descifrar	Descubrir
3	ABSTENCION	20	Descifrar	Descubrir
0	NO VOTANTE	0	Descifrar	Descubrir
		TOTAL:		
		84		

Anexo 9
Reunión virtual con delegados

