

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA DE SISTEMAS INFORMATICOS



**SISTEMA INFORMÁTICO PARA EL APOYO DE REUNIONES
Y CONTROL DE ACUERDOS DE LA ASAMBLEA GENERAL
UNIVERSITARIA DE LA UNIVERSIDAD DE EL SALVADOR,
SIARCA_AGU_UES**

PRESENTADO POR:

WENDY CAROLINA CRIOLLO HERNÁNDEZ

JAIME FERMÍN DONADO IRAHETA

JONATÁN BENJAMÍN LÓPEZ HENRÍQUEZ

CARLOS ALBERTO NOYOLA SÁNCHEZ

PARA OPTAR AL TÍTULO DE:

INGENIERO(A) DE SISTEMAS INFORMATICOS

CIUDAD UNIVERSITARIA, ABRIL DE 2018

UNIVERSIDAD DE EL SALVADOR

RECTOR:

MSC. ROGER ARMANDO ARIAS ALVARADO

SECRETARIO GENERAL:

MSc. CRISTOBAL HERNAN RIOS BENITEZ

FACULTAD DE INGENIERIA Y ARQUITECTURA

DECANO:

ING. FRANCISCO ANTONIO ALARCON SANDOVAL

SECRETARIO:

ING. JULIO ALBERTO PORTILLO

ESCUELA DE INGENIERIA DE SISTEMAS INFORMATICOS

DIRECTOR:

ING. JOSE MARIA SANCHEZ CORNEJO

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA DE SISTEMAS INFORMATICOS

Trabajo de Graduación previo a la opción al Grado de:
INGENIERO(A) DE SISTEMAS INFORMATICOS

Título:

**SISTEMA INFORMÁTICO PARA EL APOYO DE REUNIONES
Y CONTROL DE ACUERDOS DE LA ASAMBLEA GENERAL
UNIVERSITARIA DE LA UNIVERSIDAD DE EL SALVADOR,
SIARCA_AGU_UES**

Presentado por:

WENDY CAROLINA CRIOLLO HERNÁNDEZ

JAIME FERMÍN DONADO IRAHETA

JONATÁN BENJAMÍN LÓPEZ HENRÍQUEZ

CARLOS ALBERTO NOYOLA SÁNCHEZ

Trabajo de Graduación Aprobado por:

Docente Asesor:

ING. ARNOLDO INOCENCIO RIVAS MOLINA

SAN SALVADOR, ABRIL DE 2018

Trabajo de Graduación Aprobado por:

Docente Asesor:

(FIRMA)

ING. ARNOLDO INOCENCIO RIVAS MOLINA

AGRADECIMIENTOS

A Dios por prestarme vida y brindarme oportunidades para mi superación personal.

A mis padres por todos los sacrificios hechos en búsqueda de mi desarrollo personal y profesional, por su apoyo incondicional y sobre todo, por sus consejos y palabras de aliento en todo momento.

A mi abuelito Miguel; quien en vida creyó en que su nieta lograría convertirse en una ingeniera, no sabía de qué, pero ingeniera al fin; por sus mil y un consejos en momentos de decaimiento, por sus atenciones y muestras de cariño, de quién él consideraba su “Niña Biónica”; por convertirse en mi mejor amigo y confidente, por convertirse en mi motivo de lucha y a quien desde la tierra le puedo decir: *“Lo hemos logrado Miguelito”*.

A mis amigos, por su incondicional apoyo en cada una de las etapas de mi carrera, en especial a Bea Panameño, Jona Ruiz, Marina Ramírez y Salvador Saz; por todas sus atenciones y permitirme confiarles cada una de mis dificultades, a usted Qash, infinitas gracias por su apoyo en la culminación de mi trabajo de grado.

A la ASEIAS, por brindarme el espacio para desenvolverme y formar en mí, una líder estudiantil, cuyo principal objetivo fuese contribuir al desarrollo de nuestro sector; gracias por el respaldo otorgado como representante de la Facultad a nivel institucional, así como, permitirme hacer uso del recurso informáticos e instalaciones de la Asociación para la culminación de mi trabajo de grado.

A la ASEISI, por ser mi primera casa como parte del movimiento organizado estudiantil y por haber confiado a mi persona la representación estudiantil ante la AGU, permitiendo acrecentar mi formación personal y profesional-

A mis compañeros de Trabajo de Grado, por dar lo mejor de sí para culminar una etapa de arduo trabajo y sobre todo, por soportar cada una de mis observaciones.

A los buenos docentes que tuve a lo largo de la carrera, de quienes por su dedicación no solo recibí conocimientos, sino también, experiencias de vida. Gracias por amar formar profesionales competentes.

A la Asamblea General Universitaria por permitirme en conjunto a mis compañeros, la realización de nuestro trabajo de grado, poniendo a disposición el personal, información y materiales requeridos.

A la UESE por darme la oportunidad de ser becaria de la UES; a mis Trabajadores Sociales gracias por los consejos y apoyo moral brindado.

Finalmente, y no menos importantes, a mi Alma mater por formarme para la vida.

WENDY CAROLINA CRIOLLO HERNÁNDEZ

AGRADECIMIENTOS

Primeramente, quiero agradecer a Dios por permitirme llegar hasta esta etapa de mi vida sin su ayuda yo no podría haber llegado hasta aquí.

También quiero agradecer a mis padres por apoyarme en todo momento de mis estudios y ayudarme a seguir adelante.

Agradezco también por haber conocido a todas las personas que influyeron en mi vida y me ayudaron a crecer tanto como persona y como profesional, una página no es suficiente para demostrar lo agradecido que estoy con todos Amigos, compañeros, familiares.

Quiero agradecer también a nuestro docente asesor Ing. Arnoldo Rivas el cual nos apoyó en todo momento y a los miembros de la Asamblea General Universitaria por creer en nosotros para el desarrollo de este trabajo.

Agradezco también a mis compañeros de trabajo de graduación con los cuales hemos llegado hasta este punto de nuestras vidas y desearles siempre lo mejor en cada aspecto de su vida.

Agradezco a la Universidad de El Salvador por haberme formado como profesional y a todos los profesores que se dedicaron a que aprendiéramos lo necesario para avanzar en esta carrera.

Agradezco a la fundación de ayuda para niños y ancianos por haberme ayudado durante tanto tiempo.

Jaime Fermín Donado Iraheta

AGRADECIMIENTOS

El camino hasta este punto no ha sido sencillo, pero gracias a sus aportes, su amor, su bondad y apoyo he logrado cumplir una de las metas de mi vida. Les agradezco, y hago presente mi gran afecto hacia cada uno de ustedes.

A Dios, por haberme permitido llegar hasta este punto y haberme dado salud, la sabiduría y la inteligencia para lograr esta meta, además de su infinita bondad y amor. Gracias por estar presente no solo en esta etapa tan importante de mi vida, sino en todo momento ofreciéndome lo mejor y buscando lo mejor para mi persona.

Mis padres, por ser los principales promotores de mis sueños, por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de la vida, por su incondicional apoyo perfectamente mantenido a través del tiempo, gracias por cada día confiar en mí y por haberme apoyado en todo momento, por sus consejos, sus valores, por sus ejemplos de perseverancia y constancia que me han infundido siempre, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor brindado.

A mis amigos y amigas que he tenido la dicha de conocer y con los cuales he tenido agradables momentos a lo largo de este tiempo y que me acompañaron en cada una de las materias cursadas, por su amistad, apoyo y motivación brindada. Igualmente agradecer a mi grupo de graduación con quienes comparto este triunfo, gracias por todo el tiempo invertido, el trabajo realizado, por su perseverancia y dedicación.

A nuestro docente asesor, Ing. Arnoldo Rivas, muchas gracias por guiarnos en el proceso para que este fuera realizado correctamente, gracias por el ánimo y el apoyo que siempre nos brindó.

Gracias a mi universidad, gracias por haberme permitido formarme y en ella, gracias a todas las personas que fueron partícipes de este proceso, ya que fueron ustedes los responsables de realizar su aporte, que el día de hoy se vería reflejado en la culminación de mi paso por la universidad.

Jonatán Benjamín López Henríquez

AGRADECIMIENTOS

“Principalmente a Dios que me da cada vez, un día más de vida para seguir continuando”.
Mi madre Isabel.

Mi madre, quien siempre estaba cada día para mí. Siempre pendiente con pequeños detalles cada vez que salía y regresaba a casa de estudiar, dándome ánimos cuando estaba cansado y motivándome a salir adelante y no rendirme. Tantos pequeños detalles que sería imposible plasmarlos en su totalidad en este momento. Todo su amor, el amor que solo una madre puede brindar a los que más quiere, y yo tuve la dicha de sentir.

A mi padre, Cándido, quien siempre me daba soporte para seguir adelante, a pesar que en numerosas ocasiones lo defraudaba. Tantas ocasiones en las que me decía que ya no me podría apoyar con los estudios, pero de una manera u otra siempre encontraba la manera de hacerlo; mi Padre quien siempre habla de mí con sus conocidos y amigos, lo hace diciendo “mi hijo, es ingeniero” y que, a pesar de no poder compartir mucho tiempo con él, siempre he sentido que he caminado en suelo firme gracias a él.

A mis hermanos Marielos, Marcos y Mario, que, aunque no siempre estemos de acuerdo, siempre me apoyaron en las cosas que necesitaba. Y a toda mi familia que siempre ha estado pendiente de mi progreso.

A mis amigos grandiosos, esos lazos que se forman y a pasar el tiempo siempre permanecen. A cada uno de ellos, gracias por haberme ayudado en cada etapa de la travesía de mi formación Universitaria: Juan Carlos, Rosa, Azu y Lennin. Gracias a todos.

A la ASEISI por bríndame la oportunidad de crecer con ellos.

A la ASEIAS por la oportunidad de permitirme representar a la Facultad de Ingeniería y Arquitectura ante el CSU, donde tuve la oportunidad de trabajar por los estudiantes conociendo a personas que siempre llevare en el corazón sin importar que se encuentren en otro extremo del país, haberlas conocido es un recuerdo que atesorare para siempre.

Agradecer a los docentes que me ayudaron (unos más que otros) a lo largo del camino, camino que me prepararía como profesional y con quienes, a pesar de tener nuestras diferencias, estas prácticamente desaparecían al terminar el ciclo.

A mi equipo de trabajo de grado, nuestro asesor Ing. Rivas por su apoyo. A mis amigos, con quienes compartí tanto en tan poco tiempo. Siempre peleando con Wendy, desvelándome con Jonathan a la hora de programar y Jaime por responder siempre con el trabajo sin poner un “pero”. Todos tan diferentes pero dispuestos a trabajar en equipo.

Finalmente, a gracias a la Minerva, madre de todos los Estudiantes de la Universidad. Hacia la libertad por la cultura.

Carlos Alberto Noyola Sánchez

Contenido General

INTRODUCCIÓN	1
OBJETIVOS DEL PROYECTO	3
OBJETIVO GENERAL	3
OBJETIVOS ESPECÍFICOS	3
ALCANCES	5
LIMITACIONES	7
JUSTIFICACIÓN	8
IMPORTANCIA	10
CAPÍTULO I ANÁLISIS DEL SISTEMAS	11
1.1 SITUACIÓN ACTUAL	11
1.1.1. <i>Antecedentes</i>	11
1.1.2. <i>Situación actual</i>	13
1.1.3. <i>Descripción del sistema</i>	19
1.1.4. <i>Planteamiento del problema</i>	21
1.1.5. <i>Metodología para el desarrollo del proyecto</i>	24
1.2. DETERMINACIÓN DE REQUERIMIENTOS.....	29
1.2.1. <i>Metodología de la determinación de requerimientos</i>	29
1.2.2. <i>Requerimientos técnicos</i>	29
1.2.3. <i>Requerimientos informáticos</i>	36
1.2.4. <i>Requerimientos de desarrollo</i>	37
1.2.5. <i>Requerimientos operativos</i>	38
1.2.6. <i>Diagrama de enfoque de la solución propuesta</i>	44
1.2.7. <i>Casos de uso</i>	45
CAPÍTULO II DISEÑO DE SISTEMA	92
2.1. DISEÑO DE ESTÁNDARES.....	92
2.1.1. <i>Estándares de Base de datos</i>	92
2.1.2. <i>Estándares de Pantallas</i>	94
2.1.3. <i>Estándares de Código Fuente</i>	97
2.1.4. <i>Estándares de Reportes</i>	98
2.2. DISEÑO DE BASE DE DATOS	100
2.2.1. <i>Modelo conceptual</i>	101
2.2.2. <i>Modelo lógico</i>	101
2.2.3. <i>Modelo Físico</i>	101
2.3. DISEÑO DE PANTALLAS	102
2.3.1 <i>Búsqueda documentos</i>	103
2.3.2 <i>Crear comisión</i>	104
2.3.3 <i>Administrar comisión</i>	105
2.3.4 <i>Administrar integrantes de comisión</i>	106
2.3.5 <i>Trabajo de comisión</i>	107

2.3.6	<i>Gestionar Asistencia Asambleísta</i>	108
2.4.	DISEÑO DE SEGURIDAD.....	109
2.4.1.	<i>Amenazas y Vulnerabilidades</i>	109
2.4.2.	<i>Seguridad Física</i>	111
2.4.3.	<i>Seguridad Lógica</i>	114
2.5.	DISEÑO DE ARQUITECTURA.....	117
2.5.1	<i>Arquitectura Cliente – Servidor</i>	118
2.5.2	<i>Arquitectura de la aplicación</i>	119
2.6.	DISEÑO DE PRUEBAS	121
2.6.1	<i>Pruebas Unitarias</i>	121
2.6.2	<i>Pruebas integrales</i>	122
CAPÍTULO III CONSTRUCCIÓN.....		129
3.1	INICIO DE SESIÓN	129
3.1.1	<i>Modelo</i>	129
3.1.2	<i>Vista</i>	133
3.1.3	<i>Controlador</i>	135
3.2	CREAR USUARIO	136
3.2.1	<i>Modelo</i>	136
3.2.2	<i>Vista</i>	136
3.1.2	<i>Controlador</i>	137
CAPÍTULO IV PRUEBAS.....		139
4.1	TRAZABILIDAD DE CASOS DE PRUEBAS – REQUISITOS.....	139
CAPÍTULO V DOCUMENTACIÓN		140
5.1	MANUAL DE USUARIO	140
5.2	MANUAL DE INSTALACIÓN Y DESINSTALACIÓN	140
5.3	MANUAL TÉCNICO.....	140
CAPÍTULO VI PLAN DE IMPLEMENTACIÓN		142
INTRODUCCIÓN.....		142
	<i>Propósito</i>	142
	<i>Alcance</i>	143
	<i>Definiciones, siglas y abreviaturas</i>	143
	<i>Referencias</i>	144
	<i>Visión general</i>	144
PLANIFICACIÓN DE LA IMPLANTACIÓN.....		144
	<i>Responsabilidades</i>	146
	<i>Cronograma</i>	148
RECURSOS		153
	<i>Hardware</i>	153
	<i>A responsabilidad de la Asamblea General Universitaria</i>	153
CAPACITACIÓN		155
ANEXOS (PLAN DE IMPLEMENTACIÓN).....		158
CONCLUSIONES.....		160

RECOMENDACIONES	162
BIBLIOGRAFÍA	163
GLOSARIO	166
ANEXOS	169
A) SITIO WEB AGU.....	169
B) ACUERDO DE AUTORIZACIÓN.....	170
C) CARTA COMPROMISO.....	171
D) ASIGNACIÓN DE DIETAS.....	173
E) CUESTIONARIO ANÁLISIS Y DISEÑO	174
F) PLANTILLA WEB SIARCA_AGU_UES	175
G) CARTA DE ACEPTACIÓN DE REQUERIMIENTOS	176
H) HERRAMIENTAS DE RECOLECCIÓN DE INFORMACIÓN	177
<i>i. Entrevista</i>	177
<i>ii. Encuesta</i>	180
I) ALMACENAMIENTO DE ACTAS.....	183
J) FORMATO DE SOLICITUD DE USO DE LICENCIA.....	184
K) MAPA DE NAVEGACIÓN DEL SITIO	185

Índice de tablas

TABLA 1. CONDICIONES PARA LA ASIGNACIÓN DE DIETAS FUENTE: ELABORACIÓN PROPIA, REGLAMENTO INTERNO AGU, (UNIVERSIDAD DE EL SALVADOR, 2013),	18
TABLA 2. ESPECIFICACIONES DE HARDWARE, REQUERIMIENTOS TÉCNICOS FUENTE: ELABORACIÓN PROPIA, (CRIOLLO HERNÁNDEZ, DONADO IRAHETA, LÓPEZ HENRIQUEZ, & NOYOLA SÁNCHEZ, 2017)	29
TABLA 3. ESPECIFICACIONES DE HARDWARE DEL SERVIDOR DE DESARROLLO FUENTE: ELABORACIÓN PROPIA, (CRIOLLO HERNÁNDEZ, DONADO IRAHETA, LÓPEZ HENRIQUEZ, & NOYOLA SÁNCHEZ, 2017)	30
TABLA 4. ELEMENTOS CONSIDERADOS PARA REQUERIMIENTOS TÉCNICOS, SOFTWARE FUENTE: ELABORACIÓN PROPIA, (CRIOLLO HERNÁNDEZ, DONADO IRAHETA, LÓPEZ HENRIQUEZ, & NOYOLA SÁNCHEZ, 2017)	36
TABLA 5: EQUIPO DE DESARROLLO FUENTE: (CRIOLLO HERNÁNDEZ, DONADO IRAHETA, LÓPEZ HENRIQUEZ, & NOYOLA SÁNCHEZ, 2017)	37
TABLA 6: HARDWARE DE SERVIDOR DE DESARROLLO FUENTE: (CRIOLLO HERNÁNDEZ, DONADO IRAHETA, LÓPEZ HENRIQUEZ, & NOYOLA SÁNCHEZ, 2017)	37
TABLA 7. DESCRIPCIÓN DEL ADMINISTRADOR DEL SISTEMA FUENTE: ELABORACIÓN PROPIA.	42
TABLA 8. LISTA ACTOR-OBJETIVO.....	46
TABLA 9. FORMATO DE ESPECIFICACIÓN DE CASO DE USO FUENTE: (ALAVA, 2015)	47
TABLA 10. ESPECIFICACIÓN CU - MONITOREAR PETICIÓN FUENTE: ELABORACIÓN PROPIA	48
TABLA 11. ESPECIFICACIÓN CU - AGREGAR COMISIÓN FUENTE: ELABORACIÓN PROPIA	50
TABLA 12. ESPECIFICACIÓN CU - INCORPORAR ASAMBLEÍSTA FUENTE: ELABORACIÓN PROPIA.....	53
TABLA 13. ESPECIFICACIÓN CU - NOTIFICAR CONVOCATORIA FUENTE: ELABORACIÓN PROPIA	56
TABLA 14. ESPECIFICACIÓN CU - GESTIONAR COMISIÓN FUENTE: ELABORACIÓN PROPIA	58
TABLA 15. ESPECIFICACIÓN CU – REGISTRAR PETICIONES FUENTE: ELABORACIÓN PROPIA	59
TABLA 16. ESPECIFICACIÓN CU - GESTIONAR PERFILES FUENTE: ELABORACIÓN PROPIA.....	61
TABLA 17. ESPECIFICACIÓN CU - GESTIONAR USUARIO FUENTE: ELABORACIÓN PROPIA.....	63
TABLA 18. ESPECIFICACIÓN CU - REGISTRAR ASAMBLEÍSTA FUENTE: ELABORACIÓN PROPIA	65
TABLA 19. ESPECIFICACIÓN CU - GESTIONAR AGENDA FUENTE: ELABORACIÓN PROPIA	67
TABLA 20. ESPECIFICACIÓN CU - GESTIONAR ASISTENCIA DE ASAMBLEÍSTA FUENTE: ELABORACIÓN PROPIA	69
TABLA 21. ESPECIFICACIÓN CU - ASIGNAR PERMISOS FUENTE: ELABORACIÓN PROPIA	71
TABLA 22. ESPECIFICACIÓN CU -ADMINISTRAR SESIONES PLENARIAS FUENTE: ELABORACIÓN PROPIA.	72
TABLA 23. ESPECIFICACIÓN CU - GESTIONAR PUNTO FUENTE: ELABORACIÓN PROPIA	73
TABLA 24. ESPECIFICACIÓN CU - REESTRUCTURAR AGENDA FUENTE: ELABORACIÓN PROPIA.....	75
TABLA 25. ESPECIFICACIÓN CU - GENERAR REPORTE FUENTE: ELABORACIÓN PROPIA.....	76
TABLA 26. ESPECIFICACIÓN CU - INGRESAR DOCUMENTOS FUENTE: ELABORACIÓN PROPIA.....	78
TABLA 27. ESPECIFICACIÓN CU -ELABORAR INFORME COMISIÓN FUENTE: ELABORACIÓN PROPIA	80
TABLA 28. ESPECIFICACIÓN CU - CONSULTAR DOCUMENTACIÓN FUENTE: ELABORACIÓN PROPIA.....	81
TABLA 29. ESPECIFICACIÓN CU - MODIFICAR PLANTILLA FUENTE: ELABORACIÓN PROPIA	83
TABLA 30. ESPECIFICACIÓN CU - CREAR PERÍODO FUENTE: ELABORACIÓN PROPIA	86
TABLA 31. ESPECIFICACIÓN CU - ESTABLECER PARÁMETROS FUENTE: ELABORACIÓN PROPIA.....	89
TABLA 32. ESTÁNDAR DE TABLAS PARA BASE DE DATOS FUENTE: ELABORACIÓN PROPIA, (OTWELL, 2017) .	93
TABLA 33. ESPECIFICACIONES GENERALES DE DISEÑO FUENTE: ELABORACIÓN PROPIA.....	96
TABLA 34. ELEMENTOS DE PANTALLA. FUENTE: ELABORACIÓN PROPIA, BALSAMIQ MOCKUPS.....	102
TABLA 35. DEFINICIÓN DE VULNERABILIDADES FUENTE: ELABORACIÓN PROPIA.	110
TABLA 36. DESCRIPCIÓN DE ROLES DEL SISTEMA FUENTE: ELABORACIÓN PROPIA, SIARCA_AGU_UES. .	114
TABLA 37. FORMATO DE PRUEBAS INTEGRALES FUENTE: (JFERNANDEZ, 2012).....	122

TABLA 38. PRUEBAS INTEGRALES: PRUEBA P001 FUENTE: (JFERNANDEZ, 2012)	123
TABLA 39. PRUEBAS INTEGRALES: PRUEBA P002 FUENTE: (JFERNANDEZ, 2012)	124
TABLA 40. PRUEBAS INTEGRALES: PRUEBA P003 FUENTE: (JFERNANDEZ, 2012)	125
TABLA 41. PRUEBAS INTEGRALES: PRUEBA P004 FUENTE: (JFERNANDEZ, 2012)	126
TABLA 42. PRUEBAS INTEGRALES: PRUEBA P005 FUENTE: (JFERNANDEZ, 2012)	127
TABLA 43. PRUEBAS INTEGRALES: PRUEBA P006 FUENTE: (JFERNANDEZ, 2012)	128
TABLA 44. MATRIZ DE TRAZABILIDAD DE CASOS DE PRUEBAS FUENTE: (JUNTA DE ANDALUCÍA, 2018)	139
TABLA 45. ESPECIFICACIÓN DE MANUALES FUENTE: ELABORACIÓN PROPIA.....	141
TABLA 46. MATRIZ DE ASIGNACIÓN DE RESPONSABILIDADES FUENTE: ELABORACIÓN PROPIA	147
TABLA 47. CRONOGRAMA DEL PLAN DE IMPLEMENTACIÓN FUENTE: ELABORACIÓN PROPIA	148
TABLA 48. ESPECIFICACIONES DE HARDWARE DEL PLAN DE IMPLEMENTACIÓN FUENTE: ELABORACIÓN PROPIA.	153
TABLA 49. SOFTWARE DE APOYO PARA EL PLAN DE IMPLEMENTACIÓN FUENTE: ELABORACIÓN PROPIA.	154
TABLA 50. PLAN DE CAPACITACIÓN FUENTE: ELABORACIÓN PROPIA.....	155
TABLA 51. COSTO DE MANO DE OBRA PARA IMPLEMENTACIÓN ELABORACIÓN PROPIA.....	157
TABLA 52. COSTO DE PAPELERÍA PARA CAPACITACIÓN FUENTE: ELABORACIÓN PROPIA.....	157
TABLA 53. CRITERIOS PARA ASIGNACIÓN DE DIETAS FUENTE: ELABORACIÓN PROPIA, REGLAMENTO INTERNO AGU.....	173

Índice de Figuras

FIGURA 1. ANTECEDENTES SIARCA_AGU_UES FUENTE: ELABORACIÓN PROPIA, (ANTEPROYECTO-SIARCA_AGU_UES, 2017)	13
FIGURA 2. CONFORMACIÓN DE LA ASAMBLEA GENERAL UNIVERSITARIA FUENTE: ELABORACIÓN PROPIA, LEY ORGÁNICA UES, (LIBRO PRIMERO: RECOPIACIÓN ADMINISTRATIVO-FINANCIERO, 2013)	14
FIGURA 3. ESTRUCTURA ORGANIZATIVA AGU FUENTE: ELABORACIÓN PROPIA, REGLAMENTO INTERNO AGU, (LIBRO PRIMERO: RECOPIACIÓN ADMINISTRATIVO-FINANCIERO, 2013).....	15
FIGURA 4. PROCESO DE GENERACIÓN DE ACUERDO AGU FUENTE: ELABORACIÓN PROPIA, LOUES (UNIVERSIDAD DE EL SALVADOR, 2013) - BIZAGI MODELER.....	17
FIGURA 5. INTERACCIÓN DE USUARIOS CON SIARCA_AGU_UES FUENTE: ELABORACIÓN PROPIA, MS-VISIO	21
FIGURA 6. CICLO DE VIDA EN CASCADA FUENTE: ELABORACIÓN PROPIA, (SISTEMASVD, 2017)	24
FIGURA 7. FOREACH DE PHP FUENTE: (MENDOZA, 2016)	32
FIGURA 8. FOREACH CON BLADE. FUENTE: (MENDOZA, 2016).....	32
FIGURA 9. PLANTILLA BLADE FUENTE: (MENDOZA, 2016)	32
FIGURA 10. CATEGORÍA DE REQUERIMIENTOS OPERATIVOS FUENTE: ELABORACIÓN PROPIA, (BELTRAN SERRANO , SANCHEZ CASTRO, LANDAVERDE GARCIA, BOLAÑOS CANJURA, & MIRANDA RAMIREZ, 2017)	39
FIGURA 11. REQUERIMIENTOS OPERATIVOS DE SEGURIDAD. FUENTE: ELABORACIÓN PROPIA, (BELTRAN SERRANO , SANCHEZ CASTRO, LANDAVERDE GARCIA, BOLAÑOS CANJURA, & MIRANDA RAMIREZ, 2017)	43
FIGURA 12. ENFOQUE DE SISTEMA FUENTE: ELABORACIÓN PROPIA	44
FIGURA 13. DIAGRAMA DE CASO DE USO GENERAL FUENTE: ELABORACIÓN PROPIA, STARTUML	45
FIGURA 14 DSS- MONITOREAR PETICIÓN (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	49
FIGURA 15 DSS- MONITOREAR PETICIÓN (FLUJO ALTERNATIVO) FUENTE: ELABORACIÓN PROPIA, STARUML.....	49
FIGURA 16. DSS- AGREGAR COMISIÓN (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML.....	51
FIGURA 17. DSS- AGREGAR COMISIÓN (FLUJO ALTERNATIVO 4.A) FUENTE: ELABORACIÓN PROPIA, STARUML	51
FIGURA 18. DSS- AGREGAR COMISIÓN (FLUJO ALTERNATIVO 5.A) FUENTE: ELABORACIÓN PROPIA, STARUML	52
FIGURA 19. DSS- AGREGAR ASAMBLEÍSTA (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	54
FIGURA 20. DSS- AGREGAR ASAMBLEÍSTA (FLUJO ALTERNATIVO 6.A) FUENTE: ELABORACIÓN PROPIA, STARUML	55
FIGURA 21. DSS- NOTIFICAR CONVOCATORIA (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	57
FIGURA 22. DSS- NOTIFICAR CONVOCATORIA (FLUJO ALTERNATIVO 6.A) FUENTE: ELABORACIÓN PROPIA, STARUML	57
FIGURA 23. DSS- GESTIONAR COMISIÓN (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	58
FIGURA 24: DSS- REGISTRAR PETICIONES (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	60
FIGURA 25 DSS: GESTIONAR PERFILES (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	62
FIGURA 26 DSS: GESTIONAR PERFILES (FLUJO ALTERNATIVO) FUENTE: ELABORACIÓN PROPIA, STARUML	62
FIGURA 27 DSS: GESTIONAR USUARIO (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	64
FIGURA 28 DSS: GESTIONAR USUARIO (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	64
FIGURA 29 DSS: REGISTRAR ASAMBLEÍSTA (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	66
FIGURA 30 DSS: REGISTRAR ASAMBLEÍSTA (FLUJO ALTERNATIVO) FUENTE: ELABORACIÓN PROPIA.....	66
FIGURA 31. DSS- GESTIONAR AGENDA (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML.....	68

FIGURA 32. DSS - GESTIONAR AGENDA (FLUJO ALTERNATIVO 3.A) FUENTE: ELABORACIÓN PROPIA, STARUML	68
FIGURA 33. DSS - GESTIONAR ASISTENCIA DE ASAMBLEÍSTA (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	70
FIGURA 34. DSS - GESTIONAR ASISTENCIA DE ASAMBLEÍSTA (FLUJO ALTERNATIVO 3.A) FUENTE: ELABORACIÓN PROPIA, STARUML	70
FIGURA 35. DSS - ASIGNAR PERMISOS (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	71
FIGURA 36. DSS - ADMINISTRAR SESIONES PLENARIAS (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	72
FIGURA 37. DSS - GESTIONAR PUNTO (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	74
FIGURA 38 DSS: GENERAR REPORTE (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	77
FIGURA 39 DSS: GENERAR REPORTE (FLUJO ALTERNATIVO 5.A) FUENTE: ELABORACIÓN PROPIA, STARUML	77
FIGURA 40 DSS: INGRESAR DOCUMENTOS (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	79
FIGURA 41 DSS: INGRESAR DOCUMENTOS (FLUJO ALTERNATIVO) FUENTE: ELABORACIÓN PROPIA, STARUML	79
FIGURA 42. DSS - ELABORAR INFORME COMISIÓN (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	80
FIGURA 43. DSS - CONSULTAR DOCUMENTACIÓN (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	82
FIGURA 44. DSS - CONSULTAR DOCUMENTACIÓN (FLUJO ALTERNATIVO 4.A) FUENTE: ELABORACIÓN PROPIA, STARUML	82
FIGURA 45 DSS: MODIFICAR PLANTILLA (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	84
FIGURA 46 DSS: MODIFICAR PLANTILLA (FLUJO ALTERNATIVO) FUENTE: ELABORACIÓN PROPIA, STARUML	84
FIGURA 47 DSS: MODIFICAR PLANTILLA (FLUJO ALTERNATIVO) FUENTE: ELABORACIÓN PROPIA, STARUML	85
FIGURA 48 DSS: CREAR PERIODO (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML	87
FIGURA 49 DSS: CREAR PERIODO (FLUJO ALTERNATIVO) FUENTE: ELABORACIÓN PROPIA, STARUML	87
FIGURA 50 DSS: CREAR PERIODO (FLUJO ALTERNATIVO) FUENTE: ELABORACIÓN PROPIA, STARUML	88
FIGURA 51 DSS: ESTABLECER PARÁMETROS (FLUJO BÁSICO) FUENTE: ELABORACIÓN PROPIA, STARUML ..	90
FIGURA 52 DSS: ESTABLECER PARÁMETROS (FLUJO ALTERNATIVO) FUENTE: ELABORACIÓN PROPIA, STARUML	90
FIGURA 53 DSS: ESTABLECER PARÁMETROS (FLUJO ALTERNATIVO) FUENTE: ELABORACIÓN PROPIA, STARUML	91
FIGURA 54. DISEÑO DE ESTÁNDAR DE PANTALLAS FUENTE: ELABORACIÓN PROPIA BALSAMIQ MOCKUPS 3 ...	94
FIGURA 55. DISEÑO DE PANTALLA DE SALIDA DE DATOS FUENTE: ELABORACIÓN PROPIA	100
FIGURA 56. DISEÑO DE BASE DE DATOS FUENTE: (ALAVA, 2015)	100
FIGURA 57. DISEÑO DE BÚSQUEDA DOCUMENTOS FUENTE: ELABORACIÓN PROPIA BALSAMIQ MOCKUPS 3 ..	103
FIGURA 58. DISEÑO DE CREAR COMISIÓN FUENTE: ELABORACIÓN PROPIA BALSAMIQ MOCKUPS 3	104
FIGURA 59. DISEÑO DE ADMINISTRAR COMISIÓN FUENTE: ELABORACIÓN PROPIA BALSAMIQ MOCKUPS 3	105
FIGURA 60. DISEÑO DE ADMINISTRAR INTEGRANTES DE COMISIÓN FUENTE: ELABORACIÓN PROPIA BALSAMIQ MOCKUPS 3	106
FIGURA 61. DISEÑO DE TRABAJO DE COMISIÓN FUENTE: ELABORACIÓN PROPIA BALSAMIQ MOCKUPS 3	107
FIGURA 62. DISEÑO DE GESTIONAR ASISTENCIA DE ASAMBLEÍSTA FUENTE: ELABORACIÓN PROPIA BALSAMIQ MOCKUPS 3	108
FIGURA 63. ARQUITECTURA CLIENTE-SERVIDOR. FUENTE: ELABORACIÓN PROPIA / LUCIDCHART, (ALAVA, 2015)	119
FIGURA 64. MODELO MVC. FUENTE: ELABORACIÓN PROPIA / LUCID CHART, (COLEMAN, S.F.)	121

FIGURA 65. MODELO PARA EL INICIO DE SESIÓN FUENTE: MODELO FÍSICO SIARCA_AGU_UES	129
FIGURA 66. CÓDIGO PARA LA ESPECIFICACIÓN DE MODELO DENOMINADO "USER" FUENTE: SIARCA, \APP\USER.PHP	130
FIGURA 67. CÓDIGO PARA LA ESPECIFICACIÓN DEL MODELO DENOMINADO "PERSONA" FUENTE: SIARCA, \APP\PERSONA.PHP.....	131
FIGURA 68. CÓDIGO PARA LA ESPECIFICACIÓN DEL MODELO DENOMINADO "ROL" FUENTE: SIARCA, \APP\ROL.PHP.....	131
FIGURA 69. CÓDIGO PARA LA ESPECIFICACIÓN DEL MODELO DENOMINADO "MODULOROL" FUENTE: SIARCA, \APP\MODULOROL.PHP	132
FIGURA 70. CÓDIGO PARA LA ESPECIFICACIÓN DEL MODELO DENOMINADO "MODULO" FUENTE: SIARCA, \APP\MODULO.PHP	132
FIGURA 71. VISTA - INICIAR SESIÓN FUENTE: SIARCA_AGU_UES.....	133
FIGURA 72. CÓDIGO PARA VISTA DE INICIO DE SESIÓN "MODULO" FUENTE: SIARCA, \RESOURCE\VIEWS\LOGIN.BLADE.PHP	134
FIGURA 73. IMPORTACIONES PARA LA AUTENTICACIÓN DE USUARIOS FUENTE: PROYECTO SIARCA, \ APP\HTTP\CONTROLLERS\AUTO\ AUTHCONTROLLER.PHP.....	135
FIGURA 74. CLASE PARA LA AUTENTICACIÓN DE USUARIOS FUENTE: PROYECTO SIARCA, \ APP\HTTP\CONTROLLERS\AUTH\ AUTHCONTROLLER.PHP.....	135
FIGURA 75.MODELO PARA EL INICIO DE SESIÓN FUENTE: MODELO FÍSICO SIARCA_AGU_UES	136
FIGURA 76. VISTA - REGISTRAR USUARIO FUENTE: SIARCA_AGU_UES	137
FIGURA 77. IMPORTACIONES PARA EL REGISTRO DE USUARIOS FUENTE: PROYECTO SIARCA, \ APP\HTTP\CONTROLLERS\AUTH\ USUARIOCONTROLLER.PHP.....	137
FIGURA 78. CLASE PARA EL REGISTRO DE USUARIOS FUENTE: PROYECTO SIARCA, \ APP\HTTP\CONTROLLERS\AUTH\ USUARIOCONTROLLER.PHP.....	138
FIGURA 79. SITIO DE INFORMACIÓN AGU FUENTE: HTTP://WWW.AGU.UES.EDU.SV/	169
FIGURA 80. ACUERDO PARA AUTORIZACIÓN DE ADMINISTRACIÓN DE RECURSOS PARA LA GESTIÓN DE INFORMACIÓN AGU FUENTE: ARCHIVO ASAMBLEA GENERAL UNIVERSITARIA	170
FIGURA 81. CARTA DE COMPROMISO FUENTE: ELABORACIÓN PROPIA.....	171
FIGURA 82. CARTA COMPROMISO, FIRMAS SUSTENTANTES FUENTE: ELABORACIÓN PROPIA.....	172
FIGURA 83. VISTA REAL DE PLANTILLA ADOPTADA EN SIARCA FUENTE: SIARCA_AGU_UES	175
FIGURA 84. CARTA DE ACEPTACIÓN DE REQUERIMIENTOS PARTE 1 FUENTE: ELABORACIÓN PROPIA	176
FIGURA 85. CARTA DE ACEPTACIÓN DE REQUERIMIENTOS PARTE 2 FUENTE: ELABORACIÓN PROPIA	176
FIGURA 86. CARTA DE ACEPTACIÓN DE REQUERIMIENTOS PARTE 3 FUENTE: ELABORACIÓN PROPIA	176
FIGURA 87. ALMACENAMIENTO DE ACTAS Y ACUERDOS AGU FUENTE: ELABORACIÓN PROPIA	183

Introducción

La Asamblea General Universitaria es el “máximo organismo normativo y elector”¹ de la Universidad de El Salvador, conformada por un total de 144 miembros pertenecientes a los sectores estudiantiles, docentes y profesionales no docentes, de los cuales 72 funge como propietarios y 72 como suplentes

Debido al carácter y jerarquía de dicho organismo, dentro de la Universidad esta recibe anualmente cientos de peticiones, las cuales se someten a múltiples procesos burocráticos antes de ser resueltas, dichos procesos generan la acumulación de peticiones y por consiguiente, un bajo rendimiento en la realización de actividades administrativas.

Productos de lo antes expuesto, nace el **SISTEMA INFORMÁTICO PARA EL APOYO DE REUNIONES Y CONTROL DE ACUERDOS DE LA ASAMBLEA GENERAL UNIVERSITARIA DE LA UNIVERSIDAD DE EL SALVADOR** abreviado **SIARCA_AGU_UES**, como respuesta a la mejora de las actividades administrativas, de las cuales en el presente documento se conocerán generalidades, reflejando los alcances, limitaciones e importancia del proyecto, sin olvidar aspectos de análisis, diseño, desarrollo y uso del Sistema informático construido.

Cabe mencionar que el actual trabajo consta de seis capítulos, en los cuales se engloba la metodología y elementos utilizados para la culminación del trabajo de graduación, siendo los capítulos abordados los siguientes:

Capítulo I - Análisis del Sistema: aborda generalidades de la Asamblea General Universitaria, así como problemas y necesidades a cubrir con el sistema informático propuesto.

¹ Definición tomada de Art. 16, Ley Orgánica UES, (LIBRO PRIMERO: Recopilación Administrativo-financiero, 2013)

Capítulo II – Diseño del Sistema: presenta las líneas bases seguidas para el desarrollo del Sistema informáticos, así como los conocimientos mínimos requeridos para la realización de modificaciones futuras al SIARCA.

Capítulo III- Construcción: muestra aspectos fundamentales del desarrollo, plasmando un ejemplo de la funcionalidad de uno de los requerimientos definidos en la etapa de análisis y diseño.

Capítulo IV-Documentación: dicho apartado, muestra las referencias de los documentos que se entregan conjunto al software desarrollado, a fin de una buena instalación, modificación y uso.

Capítulo V- Pruebas: apartado dedicado para especificar parte de las pruebas realizadas al sistema informático, así como, el impacto que los casos de pruebas diseñados tienen con los requerimientos establecidos.

Capítulo VI-Plan de implementación: Dicho capítulo muestra los indicadores a considerar para la implementación y uso del sistema informáticos, desde el punto de vista técnico y operativo.

Objetivos del proyecto

Objetivo general

- ✚ Desarrollar un sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador que permita la optimización de tiempo y recursos, empleados por el recurso humano involucrado en las actividades a cargo del organismo antes mencionado, contribuyendo además en la mejora de su ejecución presupuestaria.

Objetivos específicos

- ✓ Dar a conocer los alcances del sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador, a fin de esclarecer hasta donde cubre la aplicación desarrollada.
- ✓ Indicar las limitaciones con las que se enfrentó el desarrollo del SIARCA.
- ✓ Presentar los antecedentes de la Asamblea General Universitaria relacionada con el uso de Sistemas Informáticos para el apoyo de reuniones, control de acuerdos.
- ✓ Dar a conocer el beneficio que trae consigo el desarrollo de un sistema informático para el apoyo de reuniones y control de acuerdos de la AGU, por medio de la justificación e importancia que este tiene para con la comunidad universitaria.
- ✓ Presentar los requerimientos y especificaciones establecidas en la etapa de análisis y diseño, solicitadas por el usuario de negocio, para el desarrollo del sistema informático propuesto, así como los estándares y políticas consideradas en el desarrollo de SIARCA_AGU_UES.
- ✓ Mostrar parte del código fuente del sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la

Universidad de El Salvador a fin de conocer la forma de cómo se llevó a cabo la programación del SIARCA_AGU_UES.

- ✓ Revelar el plan de pruebas seguido, para garantizar la implementación correcta del sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador.
- ✓ Presentar los manuales correspondientes para garantizar el buen funcionamiento y mantenimiento del SIARCA_AGU_UES.
- ✓ Dar a conocer el plan de implantación que permita la puesta en producción el sistema informático desarrollado.

Alcances

El sistema Informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador posee:

1. Un módulo de administración de roles, que permita, el ingreso o baja de usuarios, así como la gestión de permisos para las diferentes funcionalidades del sistema, creación de perfiles y definición de parámetros.
2. Un módulo de gestión de reuniones (sesiones plenarias y comisiones)² el cual permitirá:
 - ✓ El control de sesiones plenarias (inicio y cierre de las sesiones).
 - ✓ Gestión de asistencias y permisos a las sesiones plenarias y comisiones.
 - ✓ Administrar las agendas a tratar en las sesiones plenarias.
 - ✓ Administración del trabajo de las comisiones.
 - Convocatorias a comisiones.
 - La incorporación de asambleístas en las comisiones de la AGU.
3. Un módulo que permitirá la gestión y el almacenamiento de los acuerdos y dictámenes con actividades como:
 - ✓ Apoyo para la generación de actas, acuerdos y dictámenes de la Asamblea General Universitaria.
 - ✓ Almacenamiento de documentación generada por la AGU de forma digital.
4. Un módulo que permitirá el control y seguimiento de las solicitudes recibidas en la Asamblea General Universitaria por parte de la comunidad Universitaria, el registro de la correspondencia recibida como parte de las peticiones, así como dar seguimiento a ellas visualizando la instancia en donde se encuentran.

² Se pretende que este módulo sea escalable a reuniones como: Sesiones de Junta directivas y comisiones de trabajo.

5. Un módulo de generación de reportes para los tres niveles organizativos estratégico, táctico y operativo asociados con la gestión de reuniones y acuerdos tomados por la Asamblea General Universitaria, permitiendo la generación en pantalla o en de forma impresa según el caso.
6. Documentación de análisis, diseño, instalación, manuales de usuario, manual de desinstalación, los cuales servirán de apoyo con el sistema informático, ya que sirven de referencia si no se llegara a entender alguna funcionalidad del mismo.

En resumen, se pretende:

Desarrollar un sistema Informático funcional, libre de errores y aceptado por la Asamblea General Universitaria de la Universidad de El Salvador y generando los informes que se hayan solicitado al Equipo de trabajo en las formas indicadas (pantalla, papel o archivos).

Limitaciones

Se determinó la existencia de una sola limitante que podría impedir la culminación exitosa del proyecto, la cual consistía en:

- Posibles cambios consecutivos de la Legislación Universitaria, que involucren proceso relacionados con las actividades que se buscan automatizar por medio del Sistema informático a desarrollar.

Justificación

La idea de construir el sistema informático para apoyo a las reuniones y control de acuerdos de la Asamblea General Universitaria surge con el propósito de brindar una mejora general en el desarrollo de las actividades que ella realiza, logrando agilizar los procesos internos, permitiendo brindar un mejor servicio a la comunidad universitaria.

El SIARCA_AGU_UES además de apoyar a la Asamblea General Universitaria en la automatización de los procesos de gestión y desarrollo de reuniones de la Junta Directiva del organismo y las de sesiones plenarias; permitirá a la corporación universitaria mantenerse al tanto de donde se encuentran sus procesos, en caso de haber iniciado uno. De igual forma, este sistema contribuirá a la uniformización los procesos de creación y seguimiento de los puntos tratados en las agendas, además de propiciar un adecuado control en los asuntos recibidos y discutidos por las comisiones. En general, el Sistema mantendrá al tanto del acontecer en cuanto a reformas de leyes, procesos de legalización, procesos electorales y acuerdos generados en el desarrollo de las actividades mismas de la Asamblea General Universitaria.

Algunas de las problemáticas que atenderá el sistema informático sería: retraso en la publicación de acuerdos, acumulación de peticiones, elaboración tardía de reportes solicitados por unidades y facultades de la Universidad de El Salvador, así como, el retraso en pago de dietas.

Finalmente, al preguntarse ¿Por qué es necesario el desarrollo del Sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador? una respuesta estaría relacionada a sus beneficiarios, los cuales no solo sería los 144 miembros de Asamblea General Universitaria, sino también los empleados administrativos del organismo, la Fiscalía General Universitaria, la Unidad de Acceso a la Información, la Defensoría de los Derechos Universitarios, estudiantes, Facultades que conforman la UES y que requieren de al menos un servicio brindado por la AGU, es decir, toda la comunidad Universitaria en general.

En general, algunos de los beneficios que podría proporcionar el desarrollo del Sistema informático estarían:

- Optimización de tiempos en la evacuación de peticiones recibidas por la Asamblea General Universitaria.
- Renovación en los mecanismos de difusión de los acuerdos emitidos por el organismo.
- Eficacia por parte de la Junta Directiva de la AGU en las propuestas de agendas a tratarse en el pleno.
- Manejo eficiente de dictámenes y acuerdos vistos por el Organismo.
- Mejoramiento de la ejecución presupuestaria en cuanto al procesamiento oportuno para el pago de dietas.
- Elaboración oportuna de informes solicitados por las diversas facultades y unidades con las que tiene relación la Asamblea General Universitaria.
- Reducción de que los usuarios invierten en la consulta sobre las solicitudes realizadas, evitando la concurrencia en las oficinas administrativas de AGU.
- Estandarización de formatos utilizados para la creación de los documentos que se generaran en el sistema informático.

Cabe mencionar, que el impacto del proyecto a largo plazo puede llegar a ser muy significativo, ya que se lograría una mejor administración de los recursos que la unidad posee y permitiría aumentar la credibilidad en los procesos que la AGU trata, generando un impacto positivo en la comunidad Universitaria.

Importancia

La importancia del desarrollo del sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador radica en los siguientes elementos:

1. La creación de los roles de usuarios que permite mantener un registro de las actividades que realiza cada persona, los cambios que realizan en la aplicación, así como restricción de información, según permisos definidos.
2. Siendo que las sesiones plenarias son actividades que requieren de la presencia de personas en calidad de asambleístas, el SIARCA, permite un registro eficiente de las asistencias, lo cual se vincula directamente con el registro de dietas producidas con el porcentaje de permanencias de aquellos miembros propietarios.
3. Se llevará una correcta administración de los puntos agendados en las sesiones plenarias, permitiendo, la generación de un instrumento más completo para la generación de actas de sesiones plenarias
4. En vista de la constante concurrencia de solicitudes por parte de la corporación universitaria, con la implementación del sistema informático propuesto, se buscará la gestión eficiente de las mismas, desde el momento de ingreso; logrando con ello la transparencia del proceso de las solicitudes recibidas en la entidad de estudio.
5. Se busca agilizar el cálculo de dietas que es otorgados a los miembros que asisten a las reuniones y otras actividades afines, así como el de la renta que se genera por el pago de las mismas con el propósito de mejorar la ejecución presupuestaria de la Asamblea General Universitaria, generada por esa erogación de fondos.
6. Conocido que la generación de reportes es vital para cualquier sistema, esta potestad, permitirá generar información de acuerdo a las necesidades de los usuarios finales.

Lo anterior se considera importante debido a que el trabajo de la Asamblea General Universitaria genera una incidencia directa en toda la comunidad universitaria.

Capítulo I

Análisis del Sistemas

1.1 Situación Actual³

1.1.1. Antecedentes

Documentalmente se conoce que la primera aplicación⁴ utilizada por la Asamblea General Universitaria, de aquí en adelante AGU, data del 10 de agosto de 2014; se trata de un sitio web cuyo fin es la publicación de acuerdos e información del organismo, a fin de dar cumplimiento a lo solicitado por la Unidad de Acceso a la Información Pública de la UES⁵, con base a acuerdo No. 39/2013-2015⁶.

El sitio antes mencionado, es el que se usa actualmente, este sitio presenta una serie de dificultades en cuanto a administración y uso se refiere, además, por el espacio limitado, en él no es posible cargar gran cantidad de importación, lo que dificulta completar todas las secciones del sitio.

Otro intento de incorporar un sistema informático se da dentro de la gestión 2013-2015, donde un asambleísta les proporcionó un sistema que permitía realizar actividades relacionadas a la ejecución presupuestaria de la Asamblea General Universitaria, además de permitir la generación de planillas para el pago de dietas. El uso de ese sistema era exclusivo para la persona que fungía como técnico financiero en la unidad, no era un aplicativo oficial de la unidad ya que carecía de seguridad y había surgido

³ Actualmente se realiza el reguardo físico de la documentación en original y como en estantes como se indican en el anexo i)

⁴ Ver anexo a), Sitio web AGU: <http://www.agu.ues.edu.sv/?q=node/132>, con documento de primera publicación.

⁵ Entiéndase como: Universidad de El Salvador.

⁶ Ver anexo b), Acuerdo de autorización a presidente AGU para cumplir con lo solicitado por la Unidad de Acceso a la Información Pública de la Universidad de El Salvador.

como producto de las habilidades del creador y con el propósito de ayudar al personal administrativo de la AGU, por tal motivo, se dejó de usar⁷.

Para el año 2016, como proyecto de la asignatura “Administración de proyectos Informáticos”, estudiantes solicitan a la Asamblea General Universitaria información referente a necesidades que la unidad pudiese solventar con la incorporación de un sistema informático, naciendo de ello el *Sistema informático Para la Asignación de Dietas a Miembros de la Asamblea General Universitaria de La Universidad de El Salvador, SIAD-AGU/UES*, que se queda a nivel teórico con posibilidades de retomarse como trabajo de graduación futuro.

En febrero 2017, se aprovecha el contacto de la unidad con estudiantes de la Facultad para plantear la posibilidad de dar vida al Sistema informático actualmente retomado con el trabajo de graduación, el cual se transforma al pasar del trato de dietas como punto central, a la administración la toma de acuerdos y actividades relacionadas con reuniones del organismo de gobierno, por lo que compartida la idea con la Junta Directiva de la Asamblea General Universitaria, esta acepta el giro del sistema informático y se compromete con los estudiantes mediante la firma de una carta compromiso⁸.

En general, lo que permite contemplar *SISTEMA INFORMÁTICO PARA EL APOYO DE REUNIONES Y CONTROL DE ACUERDOS DE LA ASAMBLEA GENERAL UNIVERSITARIA DE LA UNIVERSIDAD DE EL SALVADOR*, como solución, toma como punto de partida 4 momentos que se resumen en la figura siguiente:

⁷ Retomado de (ANTEPROYECTO-SIARCA_AGU_UES, 2017)

⁸ Carta Compromiso entre las partes, mostrada en: (ANTEPROYECTO-SIARCA_AGU_UES, 2017) y anexo c) del presente documento

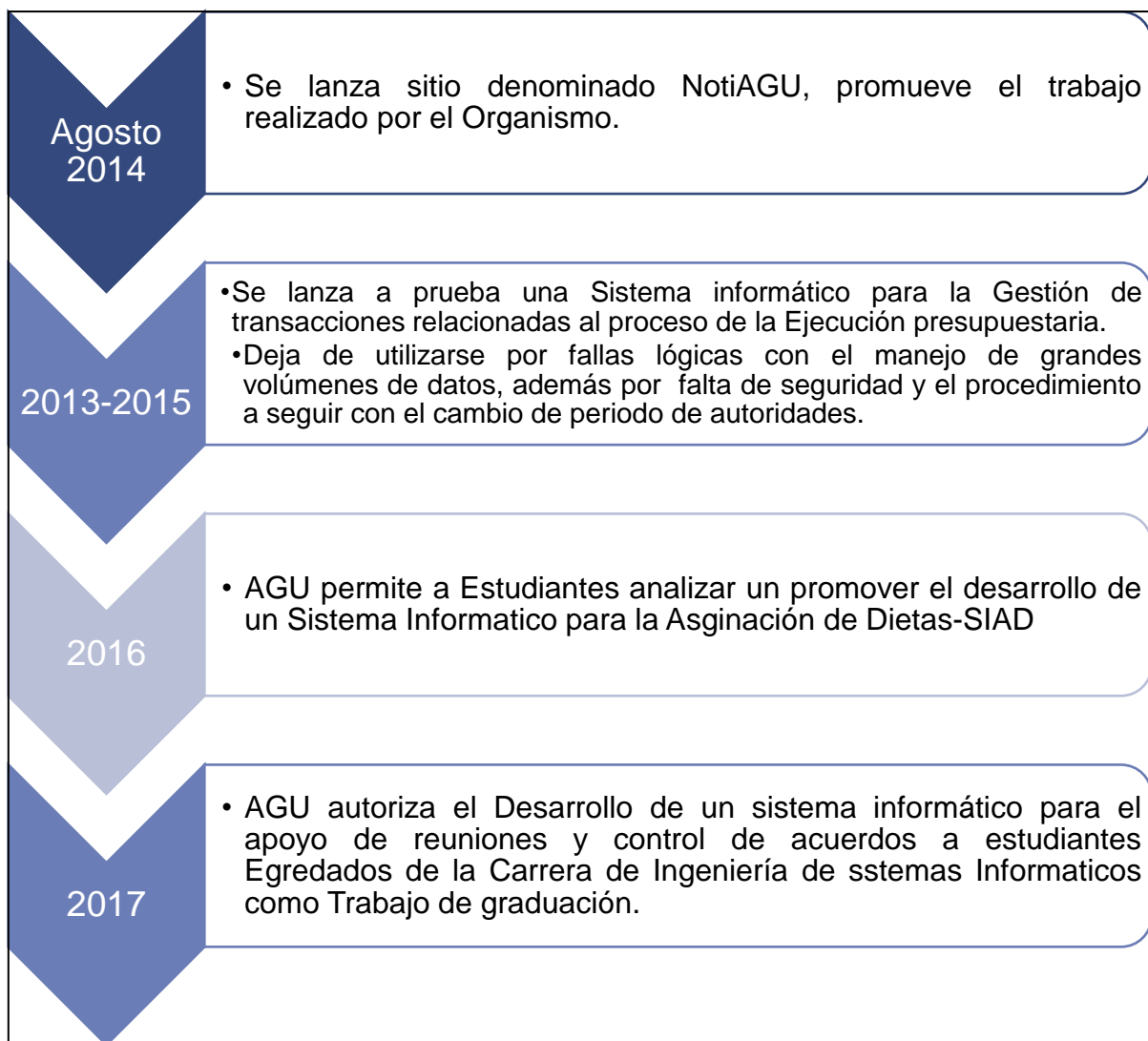


Figura 1. Antecedentes SIARCA_AGU_UES

Fuente: Elaboración propia, (ANTEPROYECTO-SIARCA_AGU_UES, 2017)

1.1.2. Situación actual

A continuación, se habla de la situación actual de la Asamblea General Universitaria, partiendo del aspecto Jurídico que regula los procesos que esta última realiza, sin antes recordar, que actualmente la AGU no cuenta con un sistema informático oficial que contribuya a la optimización de las actividades que ella realiza.

Marco legal

Las leyes que indican las atribuciones de la Asamblea General Universitaria, su composición y forma de dirigirse son:

- a) Ley Orgánica de la Universidad de El Salvador – LOUES.
- b) Reglamento de la Ley Orgánica de la Universidad de El Salvador.
- c) Reglamento Interno de la Asamblea General Universitaria.

De las Leyes anteriores, la LOUES, especifica que la Asamblea General Universitaria se encuentra conformada por la representación de 3 Sectores, así como se muestra en la Figura 2. Los cuales proponen a dos candidatos en calidad de propietarios y dos candidatos en calidad de suplentes por facultad, permitiendo integrar el Quorum del organismo por 144 personas.

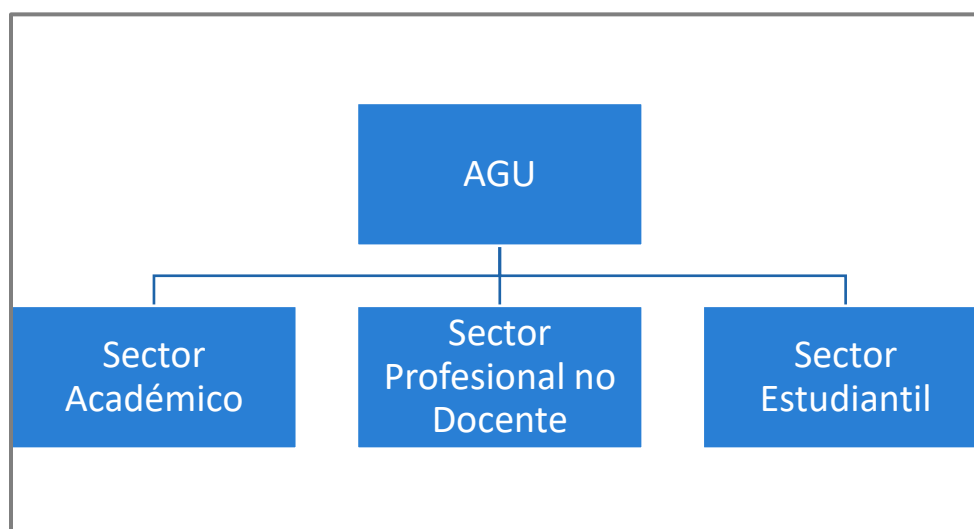


Figura 2. Conformación de la Asamblea General Universitaria
Fuente: Elaboración propia, Ley Orgánica UES, (LIBRO PRIMERO: Recopilación Administrativo-financiero, 2013)

La estructura organizativa del organismo de Gobierno estudiado tiene como máxima autoridad a los miembros de la Asamblea General Universitaria, de la cual se elige una Junta Directiva; siendo esta última la entidad encargada de administrar la unidad de negocio y gestionar lo necesario para su funcionamiento; la Junta Directiva se conforma por un Presidente, un Vicepresidente, un Secretario y dos vocales. Bajo la Junta Directiva se sitúan las comisiones permanentes que la AGU posee (Ver figura 3), además todas aquellas que se creen de forma temporal para fines específicos.

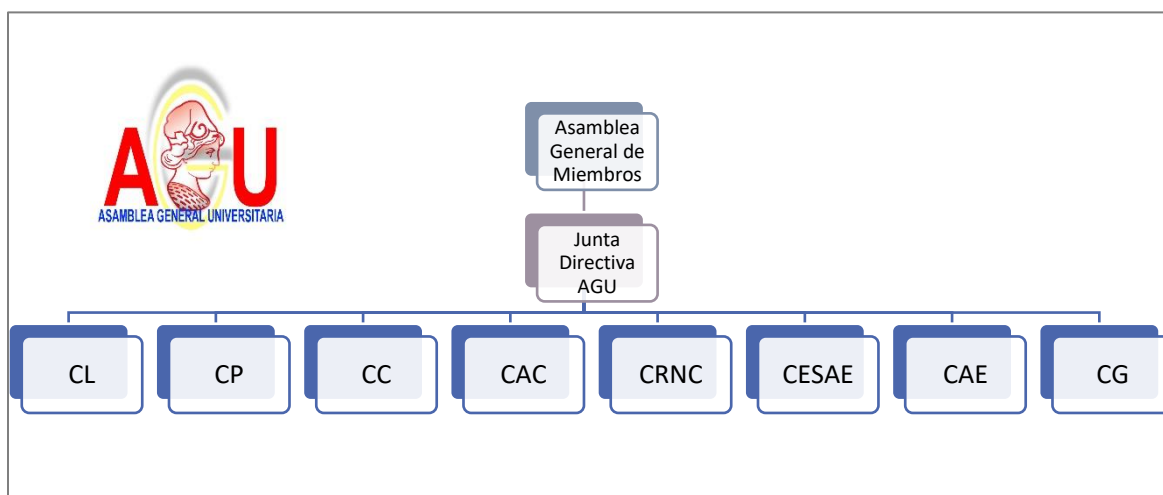


Figura 3. Estructura Organizativa AGU

Fuente: Elaboración Propia, Reglamento Interno AGU, (LIBRO PRIMERO: Recopilación Administrativo-financiero, 2013)

De la Figura 3 se tiene que las abreviaturas de las comisiones bajo la Junta Directiva de la Asamblea General Universitaria se definen de la siguiente manera:

- 1) **CL:** Comisión de Legislación.
- 2) **CP:** Comisión de Presupuestos.
- 3) **CC:** Comisión de Convenios.
- 4) **CAC:** Comisión de Arte y Cultura.
- 5) **CNC:** Realidad Nacional y Comunicaciones.
- 6) **CESAE:** Comisión de Evaluación y Seguimiento de las Autoridades Electas.
- 7) **CAE:** Comisión de Asociaciones Estudiantiles.
- 8) **CG:** Comisión de Género

Así mismo, la Asamblea General Universitaria posee dentro de sus empleados el personal siguiente:

- a) **Un Técnico Financiero:** encargado de llevar en orden los relacionado a erogación de fondos, revisión de planillas de dietas, elaboración de informes de deducción de renta.
- b) **Un Asistente Administrativo:** encargado del ingreso y control de correspondencia, así mismo, responsable de proporcionar lo requerido por las comisiones de la Asamblea general Universitaria, incluidas en esta actividad: elaboración de convocatorias, provisión de listados de asistencia y contra entrega de refrigerios, gestión de documentación vista y generada por las comisiones.
Este elemento se encarga de llevar el listado de los Asambleístas y lo relacionado a la alimentación de los mismos.
- c) **Un Profesional Universitario:** personal destacado en actividades relacionadas a la documentación de las sesiones plenarias, tales como: Elaboración de actas, transcripción de acuerdos y la gestión de agendas.
Éste profesional universitario es un elemento de apoyo para agilizar el trámite de publicación de reformas a la legislación, así como, apoyar al secretario de la AGU en documentación generada por la comisión de legislación y asociaciones estudiantiles.
- d) **Dos Colaboradores:** considerados auxiliares del profesional universitario, encargados de realizar tareas de documentación de información, elaboración de informes y apoyo en la generación de dietas conjunto al vicepresidente de la Asamblea General Universitaria.

Parte importante a señalar de la situación actual de la AGU, es el cambio de Gestión; esto debido a que fue la gestión 2015-2017 quien autorizó la realización del SIARCA_AGU_UES y es la Gestión 2017-2019 quien recibirá el producto. Sobre la gestión entrante, se puede mencionar que Presidenta en funciones fue miembros de la JD saliente, además, es con ella con quien se ha llevado a cabo la mayoría de reuniones para el establecimiento de requerimientos y solicitudes de autorización.

Descripción de procesos actuales

Generación de Acuerdos: este proceso inicia con el registro de peticiones, sigue con el tratamiento del punto en Junta Directiva, para posteriormente ser asignado o no a una comisión, la cual elabora un dictamen que será discutido por el pleno, del cual se generará el producto final, *El ACUERDO*

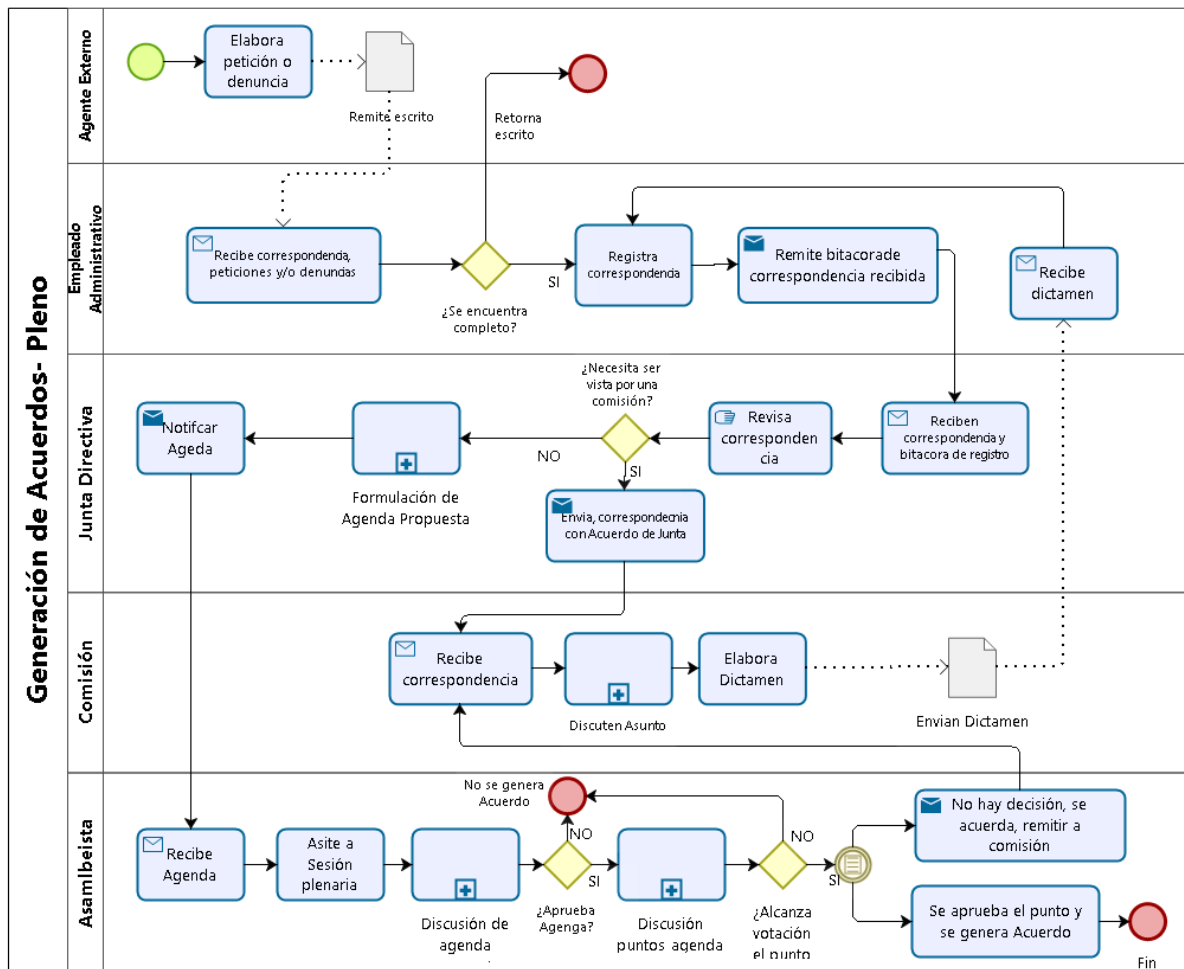


Figura 4. Proceso de generación de Acuerdo AGU

Fuente: Elaboración propia, LOUES (UNIVERSIDAD DE EL SALVADOR, 2013) - Bizagi modeler

Asignación de dietas⁹: Es un proceso adyacente a la realización de sesiones plenarias y colaboración en las comisiones, lo cual dependerá del carácter del Asambleísta que realice dicha tarea.

Del flujo anterior (Figura 4), podemos listar las actividades que generan dieta, indicando la calidad del asambleísta que lo recibe.

Este proceso obvia su ilustración, en vista que las actividades que generan dieta se describen en el proceso de generación de acuerdo, teniendo como única variación, que en proceso de asignación de dietas se involucra directamente el vicepresidente de la AGU quien se encarga de validar la permanencia de los asambleístas al 80% de las sesiones, así como corroborar la participación de los asambleístas en los dictámenes generados por la comisión, de igual manera, es el vicepresidente quien asigna dietas por acuerdos tomados en Junta Directiva. Este proceso se auxilia de un empleado de AGU en calidad de técnico financiero, quien elabora la planilla de pago de dietas una vez valida los consolidados analizados por el vicepresidente.

Tabla 1. Condiciones para la asignación de dietas

Fuente: Elaboración Propia, Reglamento Interno AGU, (UNIVERSIDAD DE EL SALVADOR, 2013),

Actividad	Rol	
	Propietario	Suplente
Participación en la resolución de asuntos trascendentales	X	X
Permanencia al 80% de la sesión	X	
Elaboración y Participación en resolución de dictámenes		X
Existencia de acuerdo de pago de dieta	X	X

⁹ Tabla de decisión referente a criterios de asignación de dietas, en anexo d) denominado: "Asignación de Dieta"

1.1.3. Descripción del sistema

El Sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria, tiene la capacidad de administrar de manera eficiente los procesos de creación y gestión de las reuniones, así como la gestión y control de acuerdos emitidos por pleno de la Asamblea General Universitaria, su Junta Directiva y las comisiones que la conforman.

Dicho sistema permite la gestión de control de usuario donde puede crear, consultar, modificar y dar de baja usuarios, roles y permisos, logrando un control de los usuarios que acceden al sistema, que modifican información de acuerdos, dictámenes y agendas.

Además, el sistema ofrece un registro de solicitudes de peticiones que la Asamblea General Universitaria recibe por parte de miembros de la comunidad universitaria, otorgando a estos último, la facilidad de conocer en cualquier momento el estado de su solicitud realizada, dando seguimiento a sus peticiones realizadas. Las personas interesadas podrán consultar en qué etapa del proceso administrativo se encuentra su petición, por medio de una sección de seguimientos de peticiones, logrando identificar, si dicha solicitud se encuentra sin ser revisada y llevar a cabo las acciones pertinentes sobre dichos casos.

Los usuarios que formen parte de la Junta directiva de la AGU, podrán realizar una agenda propuesta en base a las peticiones recibidas para ser tratadas por el pleno en la sesión plenaria, estas peticiones podrán ser dictámenes de las comisiones de la AGU o peticiones de la comunidad universitaria, dicho proceso se realizara mediante un listado de peticiones, del cual tendrá acceso la Junta Directiva de la AGU, quienes se encargaran de elegir que peticiones serán agregadas a una propuesta de agenda para la sesión plenaria. El día de la sesión será posible cambiar el orden de las peticiones, pero no se podrá agregar puntos para la agenda.

Se prevé que, a través del registro y administración de los puntos de las agendas a tratar en las sesiones plenarias, el sistema informático, automatice la generación de dietas,

El sistema es capaz de llevar un control de la asistencia de las asambleístas y trabajo de las comisiones permanentes y transitorias de la AGU, los dictámenes que presentan cada una de ellas, ayudando a los usuarios a llevar de manera más ordenada el trabajo que se realiza en cada una de las comisiones, así como a la comunidad universitaria que podrá estar al pendiente del estado de su petición.

El SIARCA_AGU_UES, cuenta con una interfaz amigable al usuario que le ofrecerá la capacidad de ingresar y almacenar los diferentes acuerdos tomados por parte de las comisiones y dictámenes en formato PDF, generando de esta manera un acceso directo por parte de los miembros de la comunidad universitaria a las decisiones tomadas por la Asamblea General Universitaria sobre acuerdos emitidos, reformas de leyes, procesos de legalización y procesos electorales, con lo cual ayudará a agilizar y facilitar el seguimiento de dichos procesos o documentos tanto para los miembros de la comunidad universitaria como a los encargados de gestionar los estados de los mismos.

Finalmente se puede decir que parte de los resultados brindados por el SIARCA_AGU_UES serían informes para los tres niveles jerárquicos de la unidad seleccionada, proporcionado desde listados de asistencias con porcentajes de permanencia y generalidades de ingreso y salida de los asambleístas, incluyendo también, listas de dictámenes vistos por comisiones en las distintas sesiones y acuerdos tomados, así como, planillas de pagos de dietas por sector y detalles de retención de rentas individuales o informes por sector, además se pretende automatizar el ingreso y salida de asambleístas de las sesiones plenarias.

En general, el funcionamiento del Sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador, seguirá la dinámica establecida en la Figura 5.

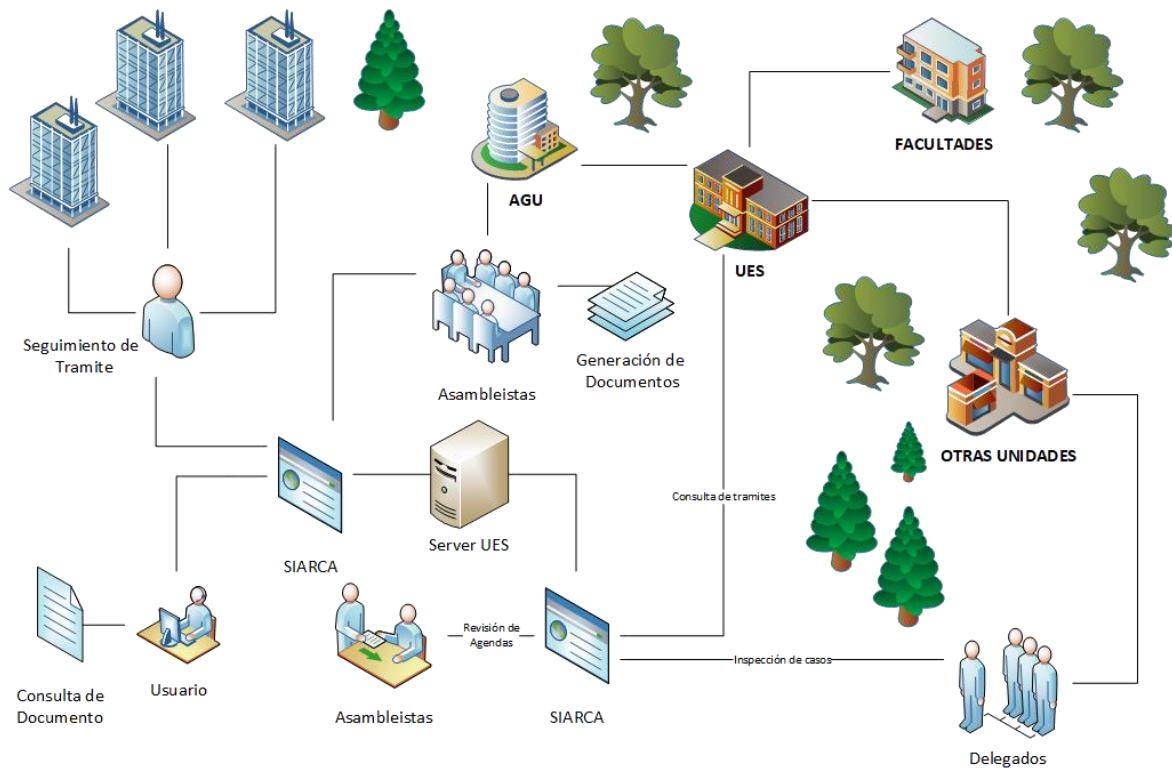


Figura 5. Interacción de Usuarios con SIARCA_AGU_UES
Fuente: Elaboración propia, Ms-Visio

1.1.4. Planteamiento del problema

Para la formulación de problemas se emplearon herramientas¹⁰ de recolección de información como:

- Observación directa en la entidad.
- Entrevistas a personal administrativo de AGU.
- Encuestas a Asambleaístas

De las cuales se determinaron los problemas siguientes:

- 1) Pago tardío de dietas, así como la emisión de comprobantes relacionados con ello, debido a que el cálculo es de forma manual para cada uno de los miembros

¹⁰ Los elementos del planteamiento del problema, se especifican a detalle en: (ANTEPROYECTO-SIARCA_AGU_UES, 2017), dicho planteamiento se obtiene de estudio realizado con herramientas especificadas en anexo h)

que cumplen con los requerimientos para poder aspirar a dicha estipendio mensual, provocando a su vez, una inadecuada ejecución presupuestaria por el retraso en el pago de dietas.

- 2) Retraso en el procesamiento de actas, acuerdos y correspondencia en general, lo que impide un manejo eficiente de las peticiones que llegan a la Asamblea General Universitaria; agregado a ello que los solicitantes desconocen el estado de las peticiones realizadas por él, a menos que se acerquen personalmente a consultar como se encuentra su petición.
- 3) Inexistencia de registro ordenado sobre los temas o peticiones a discutir en las sesiones plenarias.
- 4) Incumplimiento de tiempos para responder a peticiones de diversas unidades de la UES.

Lo anterior, por causas como:

- a) Falta de mecanismos de seguimiento del estatus sobre los documentos o peticiones que se incorporan en las reuniones.
- b) Condiciones en que se manejan los documentos en la Asamblea por ser principalmente de forma física, puede conducir al extravío de ellos debido a que pasan por diferentes manos para su procesamiento; los documentos pueden dañarse si se lleva un incorrecto almacenamiento y uso.
- c) No existe un método automático con el cual recolectar y manejar el porcentaje de asistencia y participación de los asambleístas a las sesiones plenarias, lo cual conlleva a realizar el cálculo de la dieta de manera manual y puede ocasionar equivocaciones por el personal que maneja estos datos.
- d) No existe un método automático para el cálculo de la renta que se genera por asambleísta y por sector, lo cual conlleva a realizar dicho cálculo de manera manual y puede ocasionar equivocaciones por el personal que maneja estos datos.

Por lo que en general surge la interrogante:

“¿De qué manera el sistema informático para el apoyo a las reuniones y control de acuerdos contribuirá en el desarrollo de las actividades de la Asamblea General

Universitaria y permitirá un mejor desarrollo y ejecución sobre los servicios que realiza y los recursos con los que cuenta? “

Para lo cual será necesario afrontar interrogantes específicas como:

- 1) ¿De qué manera la formulación del problema nos ayudará a identificar los problemas más relevantes que podemos resolver con la implementación del sistema SIARCA?
- 2) ¿Cómo afecta el retraso del cálculo de la dieta a los demás procesos que se realizan en la Asamblea General Universitaria?
- 3) ¿De qué manera afectan a los usuarios la consulta de los estados de las peticiones que ingresaron en la Asamblea General Universitaria?
- 4) ¿Cómo realizar la descripción del sistema de modo que los usuarios entiendan los beneficios que este les traerá su creación e implementación?
- 5) ¿Cómo lograr una total unificación de las herramientas que se utilizaran para la creación del sistema que ayudara a las actividades de la Asamblea General Universitaria?
- 6) ¿De qué forma lograr que los usuarios comprendan el completo funcionamiento y beneficios que conlleva la creación de un sistema para el apoyo de sus actividades diarias?
- 7) ¿Cómo concluir el correcto funcionamiento y puesta en marcha en el área operativa de la Asamblea General Universitaria?

Las estrategias¹¹ de las que se harán uso para dar solución a las interrogantes anteriores serían:

- FO: Fortalezas/Oportunidades
 - ✓ Aprovechamiento de los recursos que invertirá la AGU para la implementación del sistema informático SIARCA_AGU_UES
- DO: Debilidades/Oportunidades

¹¹ Estrategias elaboradas con base a FODA, que se especifica en: (ANTEPROYECTO-SIARCA_AGU_UES, 2017)

- ✓ Desarrollar un sistema informático que apoye a múltiples procesos y personas en las labores diarias de la Asamblea General Universitaria.

Capacitar al personal para el uso del sistema informático para el apoyo de reuniones y control de acuerdos de la AGU.

1.1.5. Metodología para el desarrollo del proyecto

Para el desarrollo de este proyecto se adoptó el modelo de ciclo de vida en cascada¹² que indica que el desarrollo de software se debe realizar siguiendo una secuencia de fases las cuales tienen un conjunto de metas bien definidas.

La adopción de este modelo incurre en la importancia de documentar cada una de las fases del proceso de desarrollo desde la etapa de análisis hasta el plan de implementación. La naturaleza del proyecto dispone los recursos y el tiempo para el uso de este modelo.

Las etapas que contempla el modelo seleccionado se definen en la Figura siguiente:

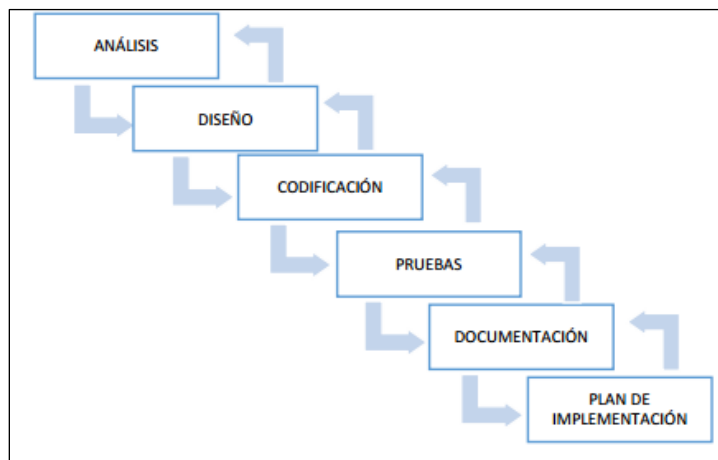


Figura 6. Ciclo de vida en cascada
Fuente: Elaboración Propia, (sistemasvd, 2017)

1.1.5.1. Análisis

Esta etapa desarrollada en el presente documento, la cual es dividida en las dos sub etapas siguientes:

¹² Criterio de elección establecido en: (ANTEPROYECTO-SIARCA_AGU_UES, 2017)

1.1.5.1.1. Análisis de la situación actual

Se analiza a detalle lo que está ocurriendo en la unidad de negocio (Asamblea General Universitaria de la Universidad de El Salvador), con el objetivo de conocer el problema y las oportunidades. Las personas involucradas en esta fase son los usuarios, analistas y los administradores de sistemas de la unidad.

Lo importante de esta etapa es la familiarización con los procesos de la Asamblea General Universitaria por parte del equipo de trabajo, conociendo quienes y como lo realizan. Se pretende conocer los problemas y los puntos de mejora mediante la aplicación de un sistema informático.

Para la determinación de los problemas se hace uso de las herramientas de recolección de datos: entrevista, observación y recolección de documentos¹³.

- a) **Entrevista.**
- b) **Observación Directa:**
- c) **Recopilación de documentos**

1.1.5.1.2. Determinación de requerimientos

Esta etapa supone encontrar y registrar lo que se necesita realmente que el sistema haga, de manera que sea fácil de entender para el cliente y el equipo de trabajo, por medio de la realización de todas las reuniones necesarias para determinar las necesidades (requerimientos), de los usuarios del sistema.

Además de lo anterior, se determinan los requisitos funcionales, especificando otros requisitos relacionados con aspectos como atributos de calidad, restricciones de software y hardware, rendimiento, seguridad, etc.

Se considera que la determinación de requerimientos, es de suma importancia debido a que una mala interpretación de ellos, puede provocar atrasos en las etapas subsiguientes del ciclo de vida; por tanto, los requerimientos deberán ser debidamente validados por el equipo de trabajo y la contraparte.

¹³ Consultar definición de herramientas en: (ANTEPROYECTO-SIARCA_AGU_UES, 2017)

Para el análisis de los requerimientos, se crearán los artefactos llamados Casos de Uso, los cuales son historias del uso de un sistema para alcanzar los objetivos. El modelo de caso de uso es un mecanismo que ayuda a mantener simple y entendible la manera en la que se comportará el sistema. Se realizará un diagrama de casos de uso haciendo uso del lenguaje UML¹⁴.

1.1.5.2. Diseño

La etapa de diseño del sistema consiste en determinar cómo el sistema cumplirá con los requerimientos que se determinaron en la fase de análisis. Se hace uso de la técnica orientada a objetos.

El artefacto UML que se crea en esta etapa es el diagrama de clases, el cual representa las especificaciones de las clases e interfaces software en una aplicación, las asociaciones, atributos, navegabilidad, dependencias y métodos entre las mismas clases.

Se establecen prototipos de las pantallas validadas por los usuarios, permitiendo la definición de estándares de las pantallas, de la estructura de los menús, comportamientos de mensajes, etc.

En esta etapa se lleva a cabo el diseño de datos., creando las bases de datos y las relaciones entre las tablas necesarias para el soporte de la aplicación; comprendiendo, además, la elaboración de los diseños conceptuales, lógicos y físicos de la base de datos.

1.1.5.3. Codificación

En esta etapa se transforma lo plasmado en el diseño en un software funcional mediante la codificación por medio de un lenguaje de programación.

El éxito de esta etapa depende en gran medida de un buen análisis y diseño. Se escriben programas para satisfacer las necesidades de otros. Comprender que necesita el cliente puede llegar a ser una tarea complicada ya que el usuario en general

¹⁴ Consultar Definición en Glosario de términos, en las secciones finales del presente documento.

no sabe expresar bien lo que necesita y requiere la amplia colaboración del programador.

1.1.5.4. Pruebas

Esta etapa, permite determinar si se cumplen los objetivos o requerimiento establecidos, colaborando, además, en la identificación de errores de lógica cometidos durante la codificación. Para su realización se hizo uso de un banco de datos ficticios pero congruentes que permitieron simular la puesta en marcha en producción.

1.1.5.4.1. Plan de pruebas unitarias

Una prueba unitaria es una forma de probar el correcto funcionamiento de un método. Esto sirve para asegurar que cada parte funcione correctamente por separado.

Estas consisten en probar una a una cada pequeña parte del sistema. Es importante hacer una prueba minuciosa de todos los escenarios posibles que se puedan dar una vez que el sistema esté implementado. Es de suma importancia verificar que los procesos de validación que se crean en la codificación estén funcionando correctamente. Una de las ventajas de las pruebas unitarias radica en su tiempo de ejecución (es menor)

1.1.5.4.2. Pruebas Integrales

Una vez realizadas las pruebas modulares (unitarias) se ejecutan las pruebas integrales, para garantizar la perfecta unión de las diferentes partes que conforman el sistema. Se busca revisar el rendimiento global del software.

Tanto en las pruebas unitarias como en la prueba integral se harán uso de un banco de datos que incluyan datos buenos y malos. Lo anterior es para comprobar que los procesos de validación de las capturas de datos funcionan correctamente.

Las pruebas de integración de software son de suma importancia para verificar que las partes del trabajo de desarrollo se comuniquen de manera correcta con las partes con las cuales interactúan, estas pruebas se realizan después de realizar las pruebas unitarias de cada módulo.

Objetivos de las pruebas de integración.

- Pruebas de interfaces
- Problemas de configuración

1.1.5.5. Documentación

1.1.5.5.1. Interna

Cada uno de los elementos de la codificación (clases, plantillas de diseño, páginas HTML, etc.) Deberán de ir documentadas. Esta parte se realizará en el momento de la codificación del sistema.

1.1.5.5.2. Externa

La documentación externa consiste en la elaboración de **manuales** que ayuden al personal de la AGU a un mejor entendimiento del sistema desarrollado. Por lo que se crearon los siguientes documentos:

Manual de usuario: Documento que servirá como guía para la utilización del software.

Elaborado de manera clara, detallando cada uno de los pasos que se deben de realizar para operar la aplicación de la forma adecuada. Se crea de forma atractiva a la vista del usuario con la cantidad necesaria de ilustraciones para garantizar un mejor entendimiento.

Manual técnico: busca guiar de forma clara al personal que le dará mantenimiento al software. Incluye todos los aspectos que se deben de tener en cuenta a la hora de realizar una modificación en el sistema.

Manual de instalación y desinstalación: Documento cuyo fin es orientar en la instalación del software y de todas las aplicaciones que este requiera (sistema gestor de base de datos, servidor de aplicaciones, entre otros). El manual detalla los procesos a seguir para instalar el sistema en varios sistemas operativos del mercado. También incluye los pasos para su posterior desinstalación.

1.1.5.6. Plan de implementación¹⁵

Se trata de un documento que especifica los pasos a seguir y las consideraciones a tomar en cuenta al momento de implementar el sistema que se desarrollado. Este plan incluye:

- Cronograma de actividades para llevar a cabo la implementación.
- Presupuesto de la implementación.
- Recursos necesarios: hardware, software y recurso humano.
- Plan de capacitación.

1.2. Determinación de requerimientos

1.2.1. Metodología de la determinación de requerimientos

Para la determinación de requerimientos, se siguió lo descrito en la sección 1.1.5.1.1 y 1.1.5.1.2, del presente capítulo.

De los cual se obtuvieron los requerimientos descritos en las secciones subsiguiente

1.2.2. Requerimientos técnicos

Para el desarrollo de SIARCA_AGU_UES, el equipo de desarrollo identificó atributos, capacidades y características del software y hardware que permitirían la culminación exitosa del sistema informático siendo ellos los descritos en las secciones siguientes.

1.2.2.1. Requerimientos de Hardware.

Las características del equipo informático con las que contó el equipo de desarrollo se definen en la tabla 2.

Tabla 2. Especificaciones de Hardware, Requerimientos Técnicos
Fuente: Elaboración propia, (Criollo Hernández, Donado Iraheta, López Henriquez, & Noyola Sánchez, 2017)

CARACTERÍSTICAS	ESPECIFICACIONES MÍNIMAS
PROCESADOR	Intel Pentium, 2 Núcleos y 3 MB de Cache
MEMORIA RAM	4 GB
SISTEMA OPERATIVO	Windows 10 de 64 bits
DISCO DURO	500 GB
PANTALLA	14" o mas

¹⁵ Se brinda para que la Asamblea General Universitaria se haga cargo de la implantación del SIARCA.

CARACTERÍSTICAS	ESPECIFICACIONES MÍNIMAS
PERIFÉRICOS	Teclado, Mouse, Tarjeta de red Fast Ethernet 10/100Mbps

Las especificaciones del equipo informático empleado como servidor poseían las características descritas a continuación:

Tabla 3. Especificaciones de Hardware del Servidor de Desarrollo
Fuente: Elaboración propia, (Criollo Hernández, Donado Iraheta, López Henriquez, & Noyola Sánchez, 2017)

CARACTERÍSTICAS	ESPECIFICACIONES MÍNIMAS
PROCESADOR	Intel Core i3-4170, 2 Núcleos y 3 MB de Cache
MEMORIA RAM	8 GB
DISCO DURA	1 TB
PERIFÉRICOS	Teclado, Mouse, Tarjeta de red Fast Ethernet 10/100Mbps

1.2.2.2. *Requerimientos de software.*

En el punto anterior se dio a conocer el hardware a utilizar para el desarrollo del sistema informático, a continuación, se detalla el software a utilizar para el mismo fin, los cuales se presentan a continuación:

1.2.2.2.1. *Lenguaje de Programación*

El lenguaje de programación adoptado para el desarrollo del SIARCA_AGU_UES, es PHP V 5.6, debido a la experiencia de uso del equipo de trabajo y la compatibilidad de herramientas utilizadas en la UES por la DTI.

Otra de las razones por las que se adopta PHP V 5.6, es debido al nivel de popularidad, lo cual permite una amplia comunidad de desarrolladores y de forma consecuente, el acceso a extensa documentación (manuales, tutoriales, talleres, foros, ejemplos de código, etc.)

PHP cuenta con una sintaxis amigable, de fácil entendimiento y uso, además de una enorme cantidad de librerías especializadas, por lo que solo basta con comprender su sintaxis para generar grandes resultados y obtener aplicaciones web muy sólidas con funciones realizadas por medio de las librerías como: conexiones a diferentes bases de datos, web services, parseo de XML, envío de email, generación de PDF, imágenes, etc. PHP cuenta con una alta gama de framework, los cuales fomentan el uso de un

modelo o un patrón de diseño, el uso de buenas prácticas que generan un código de calidad y fácilmente mantenible, lo cual incrementa exponencialmente la productividad de los desarrolladores.

1.2.2.2.2. Framework

Desarrollar un proyecto conlleva muchos desafíos, independientemente de su naturaleza, en el caso de un proyecto que implica el desarrollo de una aplicación web, nos podemos encontrar con desafíos comunes, tales como el manejo de rutas, acceso a base de datos, organización de la estructura del proyecto, seguridad, etc., además de las características y la lógica del negocio que el sistema debe modelar, razón por la cual el uso de un framework ayuda a cubrir dichos desafíos.

En cuanto a la elección de un framework que trabajase con PHP, el equipo de trabajo determinó que el indicado sería Laravel versión 5.4.

Las razones, por las que se define por Laravel, obedecen a las potestades siguientes:

- 1. Uso de patrones de diseño:** los cuales ayudan a encontrar soluciones efectivas a problemas comunes, presentando esquemas para la definición de estructuras para la construcción de aplicaciones. Laravel hace uso del patrón modelo vista controlador (MVC), mejor conocido como modelo en tres capas: Modelo (capa encargada de gestionar los datos), Controlador (capa encargada de la lógica del negocio) y Vista (capa encargada de la presentación de los datos).
- 2. Uso de ORM:** Laravel hace uso de un ORM (Object Relationship Mapping) llamado Eloquent, haciendo el manejo de los datos menos complejo. Eloquent permite una interacción con las bases de datos es totalmente orientada a objetos, siendo compatible con la gran mayoría de las bases de datos del mercado actual y facilitando la migración de nuestros datos de una forma fácil y segura. Otro punto es que favorable es que prescinde del uso de SQL en el código.
- 3. Composer:** Tecnología que permite incorporar paquetes externos a determinada aplicación, por medio del administrador de dependencias; por lo

que ya no es necesario realizarlo de forma manual (descargar una librería, descomprimir y colocar en un proyecto).

- 4. Templates Blade:** Laravel incluye un sistema de procesamiento de plantillas llamado Blade. Este sistema de plantillas favorece un código mucho más limpio en las Vistas, además de incluir un sistema de Caché que lo hace mucho más rápido. El sistema Blade de Laravel, permite una sintaxis mucho más reducida en su escritura. Por ejemplo, un foreach con PHP sería de la siguiente manera:

```
1 <?php foreach($items as $item:?)  
2 <li><?php $item;?>  
3 <?endforeach?>
```

Figura 7. ForEach de PHP
Fuente: (Mendoza, 2016)

En cambio, haciendo uso de Blade sería de la siguiente manera:

```
1 @foreach($item as item)  
2 <li>{{ $item }}</li>  
3 @endforeach
```

Figura 8. ForEach con Blade.
Fuente: (Mendoza, 2016)

Otra del sistema de plantillas que emplea Laravel, es tener una mejor estructuración de nuestras vistas, por medio de la extensión de vistas a través de vistas usando codeblocks. Un ejemplo para una plantilla HTML5 de Blade, sería el descrito en la Figura 9.

- 5. Amplia comunidad y extensa documentación:** Laravel, al contar con una amplia comunidad activa de desarrolladores, además, de contar con una

```
<!DOCTYPE html>  
<html lang="es">  
  <head>  
    <meta charset="UTF-8">  
    <title>@yield('titulo')</title>  
  </head>  
  <body>  
    @yield('navegacion')  
  </body>  
</html>
```

Figura 9. Plantilla Blade
Fuente: (Mendoza, 2016)

enorme cantidad de documentación, por parte de su sitio oficial.

- 6. Robusto sistema de Routing:** Laravel permite la creación de rutas de manera muy simple, admitiendo la agrupación de ellas para la colocación de filtros de condiciones de acceso a cumplir, favoreciendo la seguridad de la aplicación.

1.2.2.2.3. Editor de Código Fuente

El software por el que se optó para llevar a cabo la edición del código fuente, fue Sublime Text, basado en las siguientes características:

1. **Minimapa:** provee una previsualización de la estructura del código, una funcionalidad bastante útil para desplazarse por el archivo.
2. **Multi-Selección:** Hace una selección múltiple de un término por diferentes partes del archivo.
3. **Multi Cursor:** Creación de cursores para escribir texto de forma arbitraria en diferentes posiciones del archivo.
4. **Multi-Layout:** Brinda la posibilidad de división de hasta cuatro ventanas verticales o cuatro ventanas en cuadrícula.
5. **Soporte nativo para infinidad de lenguajes:** Soporta de forma nativa 43 lenguajes de programación y texto plano, lo cual incluye PHP y sintaxis SQL.
6. **Búsqueda Dinámica:** Se puede hacer búsqueda de expresiones regulares o por archivos, proyectos, directorios, una conjunción de ellos o todo a la vez.
7. **Auto completado y marcado de llaves:** Se puede ir a la llave que cierra o abre un bloque de una forma sencilla.
8. **Soporte de Plugins:** multitud de plugins que incrementan la funcionalidad de este.
9. **Configuración total de Keybindings:** Todas las teclas pueden ser sobrescritas a nuestro gusto.
10. **Coloreado y envoltura de sintaxis:** Resaltado de las expresiones propias de la sintaxis del lenguaje de programación usado para facilitar su lectura.
11. **Pestañas:** Se pueden abrir varios documentos y organizarlos en pestañas.
12. **Resaltado de paréntesis e indentación:** Cuando el usuario coloca el cursor en un paréntesis, corchete o llave, resalta ésta y el paréntesis, corchete o llave de cierre o apertura correspondiente.

1.2.2.2.4. Control de Versiones

Una de los elementos indispensables en el desarrollo de un proyecto informático, es contar con un entorno que sea un repositorio de las versiones del código fuente, para efectos del proyecto, se optó utilizar Git por medio de GitLab.

Git es un sistema distribuido de control de código fuente o SCM (Source Code Management), que permite a los miembros de un equipo de desarrollo trabajar de manera distribuida (descentralizada).

Las características que permitieron que seleccionar Git como el software para el control de versiones del código fuente del proyecto son:

1. **Facilita el trabajo colaborativo:** Distintos programadores pueden estar editando el mismo archivo, o versiones distintas del mismo archivo, y todos los cambios serán reflejados en el documento final.
2. **Reduce considerablemente los tiempos de deploy** (despliegue) de un proyecto, al subir solamente los cambios, que en Git se conoce como "diff".
3. **Permite regresar a versiones anteriores de forma sencilla y muy rápida.** En caso de haber realizado cambios negativos en un proyecto en producción, volver a la última versión estable por medio de un comando, que retrocede a su estado previo todos los cambios realizados en la última modificación. Esto puede hacerse hacia cualquier versión del proyecto, sin importar la cantidad o calidad de los cambios posteriores.
4. **Permite generar flujos de trabajo** que facilitan el desarrollo y mantenimiento de proyectos de gran tamaño.
5. **Las "branches" o ramas,** permiten trabajar con una base de código paralela al proyecto en sí, donde podemos corregir bugs o desarrollar nuevas características para el producto sin afectar el "master", pero manteniendo todas las ventajas de usar un sistema de control de versiones. Una vez que estamos contentos con nuestro "branch", podemos combinarlo con el "master" o, en lenguaje Git, hacer un "merge".
6. **Coste,** Git es un software de licencia gratuita

7. **Comunidad de desarrolladores**, Actualmente, Git es el sistema de control de versiones mayormente usado y es por ello que posee la mayor comunidad, con las ventajas que ello conlleva ya que cuando te encuentres un problema, encontrarás rápidamente la solución en la multitud de foros y chat que hay dedicados a Git.

1.2.2.2.5. Software de Modelado de UML

El software a utilizado en el proyecto para la elaboración de los diagramas UML a utilizar fue StarUML, debido a su licencia gratuita, soportando la mayoría de los diagramas que UML 2.0 ofrece, tales como: Diagramas de clases, objetos, casos de uso, componentes, despliegue, diagramas de secuencia, entre otros.

StarUML además es un software que ofrece una alta gama de funcionalidades, tales como la generación de código a partir de los diagramas y viceversa, entre los lenguajes que soporta están C++, C# y Java, también cuenta con la capacidad de exportar los diagramas generados como archivos PDF, así como exportarlos como archivos HTML y poder compartirlos de manera más fácil con otros desarrolladores y analistas.

StarUML cuenta con un repositorio en Github en el cual se pueden buscar e instalar extensiones, las cuales son OpenSource e incrementar las funcionalidades de este software.

1.2.2.2.6. Base de Datos

La base de datos a utilizada en desarrollo fue MariaDB.

La razón por la que se optó por este sistema gestor de base de datos, es por su licencia GPL, que permite su uso sin la necesidad de pago; es por ello que la Universidad de El Salvador lo emplea en sus sistemas informáticos. Se escoge MaríaDB con el fin de adaptarnos a las tecnologías de desarrollo de la UES en los servidores de la institución.

Entre las características que MariaDB ofrece son:

1. **Nuevos motores de almacenamiento más eficientes.** Aria y XtraDB vienen a reemplazar a MyISAM e InnoDB respectivamente. Cabe destacar el mayor

rendimiento de Aria, cuando recibe consultas complejas y tiene que realizar tablas temporales, éstas se cachean en memoria en vez de escribirlas en disco.

2. **Incorporación de nuevas tablas a nivel del sistema**, que ayudan en las tareas de optimización de bases de datos gracias al almacenamiento de estadísticas del servicio.
3. **El rendimiento de MariaDB es elevado** debido a que el motor Aria en particular presenta mejoras de velocidad sobre todo en consultas complejas cuando se usa el motor de almacenamiento Aria, ya que Aria cachea los datos de tablas temporales en memoria, lo que supone un rendimiento frente al uso del disco duro (que es lo que emplea MyISAM).
4. **El sistema para manejar las conexiones se ha mejorado**, ya que implementa el sistema pool-of-threads de MySQL 6.0 con el que podemos tener más de 200.000 conexiones a MariaDB.
5. Es un software de licencia gratuita, por ende, su uso no requiere de ninguna inversión económica.

En resumen, de lo antes expuesto, las tecnologías a utilizar en el desarrollo del sistema informático, son las que se muestran a continuación:

Tabla 4. Elementos considerados para requerimientos Técnicos, Software
 Fuente: Elaboración propia, (Criollo Hernández, Donado Iraheta, López Henriquez, & Noyola Sánchez, 2017)

SOFTWARE	
LENGUAJE DE PROGRAMACIÓN	PHP V 5.6.
FRAMEWORK	Laravel v5.4
EDITOR CÓDIGO FUENTE	Sublime Text
CONTROL DE VERSIONES	Git v2.13.1
SOFTWARE DE MODELADO UML	StarUML v2.8.0
BASE DE DATOS	MariaDB v10.2.6

1.2.3. Requerimientos informáticos



Véase en documento “Manual Técnico.pdf”, sección 7.1.1., adjunto en CD: /Manuales/Manual Técnico.pdf

1.2.4. Requerimientos de desarrollo

Equipo utilizado para desarrollo:

El equipo que se utilizó para el desarrollo del sistema SIARCA contaba con las siguientes características:

Tabla 5: Equipo de desarrollo

Fuente: (Criollo Hernández, Donado Iraheta, López Henríquez, & Noyola Sánchez, 2017)

HARDWARE	
CANTIDAD	Especificación
1	Desktop Procesador: Intel Core i3 4170 RAM: 8GB DDR3 Disco duro: 500 GB Adicional: Intel Graphics 3000, Ethernet 10/100/1000
1	Desktop Procesador: Intel Pentium, 3.40GHz RAM: 2GB DDR3 Disco duro: 500 GB
1	Desktop Procesador: Intel Pentium, 3.1GHZ zoket 1150 RAM: 6GB DDR3 Disco duro: 1 TB
1	Desktop Procesador: Intel Pentium, 3.40GHz RAM: 2GB DDR3 Disco duro: 500 GB

Servidor utilizado para desarrollo:

El servidor utilizado durante el desarrollo del sistema poseía las siguientes características:

Tabla 6: Hardware de servidor de desarrollo

Fuente: (Criollo Hernández, Donado Iraheta, López Henríquez, & Noyola Sánchez, 2017)

ELEMENTO	ESPECIFICACIONES
MODELO	Computadora de escritorio genérica
PROCESADOR	Intel Core i3 4170
MEMORIA	8 GB de Memoria RAM
ALMACENAMIENTO INTERNO	1 TB
FUENTE DE ALIMENTACIÓN	1 Fuente de alimentación de 550 W CA

El software que albergaba era:

- MS Windows 10 Professional
- o Windows 7 ultimate.
- Office 2010 Profesional.
- Google Chrome.
- Mozilla Firefox.
- SublimeText.
- MariaDB.
- Apache.
- PHP.

Recurso Humano

El recurso humano técnico disponible se encontraba conformado por 4 analistas programadores que desempeñaban los siguientes cargos:

- Líder del proyecto (1).
- Desarrolladores (3).
- Miembros de la Asamblea General Universitaria.

Los conocimientos y habilidades con los cuales contaba el equipo de trabajo eran:

- Programación en plataformas OpenSource.
- Uso de herramientas OpenSource.
- Análisis y diseño de sistemas.
- Administración de proyectos informáticos.
- Administración y creación de base de datos.

1.2.5. Requerimientos operativos¹⁶

Los requerimientos operativos definidos para el buen funcionamiento del SIARCA en la unidad de negocio se agrupan tal como se muestran en la figura siguiente:

¹⁶ Retomado de: (SISTEMA INFORMÁTICO PARA LA ADMINISTRACIÓN DE BECAS DE POSTGRADO EN LA UNIVERSIDAD DE EL SALVADOR., 2017)

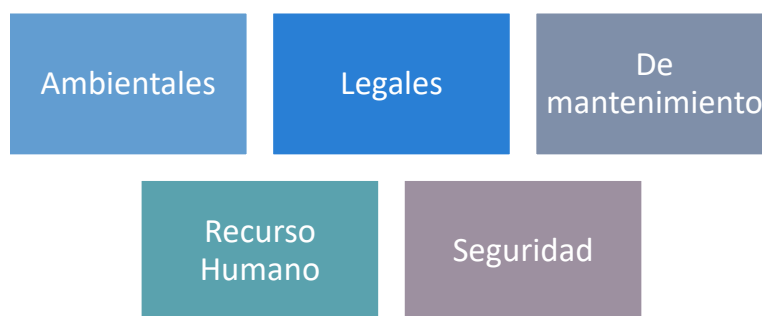


Figura 10. Categoría de Requerimientos Operativos

Fuente: Elaboración propia, (BELTRAN SERRANO , SANCHEZ CASTRO, LANDAVERDE GARCIA, BOLAÑOS CANJURA, & MIRANDA RAMIREZ, 2017)

a) Requerimientos ambientales

Los requerimientos operativos considerados para el óptimo Funcionamiento del SIARCA serán:

1. Aseguramiento del suministro Eléctrico. Considerando la estabilidad de Voltaje y su regulación por medio del uso de unidades de potencia contra cortes de corriente (UPS), con el propósito de garantizar el resguardo de la información manejada ante el corte o interrupción de la energía eléctrica.
2. Refrigeración de la zona de alojamiento de servidores.
3. Evitar el consumo de Alimentos y bebidas al operar los equipos clientes.
4. Todas las recomendadas por la Dirección de Tecnologías de la UES a efectos de garantizar el resguardo de la Información que la AGU genere.

b) Requerimientos legales

Sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador, tal como lo establecen las **Políticas Generales de Trabajo de Graduación** en su **Disposiciones Generales**, “Será propiedad de la Universidad de El salvador, y otorgada la licencia de Uso únicamente por la Facultad de Ingeniería y Arquitectura, por medio de nota Escrita dirigida a al Director de Escuela de Ingeniería de Sistemas Informáticos.”

c) **Requerimientos de Mantenimiento de equipo**

Los mantenimientos necesarios para el equipo informático que se utilice para acceder y manipular el sistema desarrollado contempla:

- ✓ **Mantenimiento Preventivo:** atendiendo de forma general y periódica los equipos de computacionales a fin de prolongar su vida útil, buscando, además, la prevención de fallas de hardware y software.

El mantenimiento preventivo debe considerar actividades como:

- Limpieza periódica de los equipos
 - Actualizado de antivirus
 - Eliminación de spyware
 - Desfragmentación de los discos duros de los equipos
 - Programación de copias de seguridad (backup) de los contenidos sensibles
 - Revisión de la configuración de la conectividad entre equipos y su acceso a Internet
- ✓ **Mantenimiento Correctivo:** se realiza para solucionar fallas que se presenten tanto en el software como en el hardware. Por lo regular, el problema se presenta inesperadamente y es que se atienda en el menor tiempo posible, para que el computador pueda volverse a usar. Entre las principales actividades de se tienen:
 - Formatear el disco duro
 - Reinstalar el sistema operativo y los programas necesarios para el funcionamiento del sistema informático.
 - Instalar y configurar los controladores apropiados para los diferentes periféricos.
 - Restauración de las bases de datos
 - Configurar y restaurar las conexiones de red
 - ✓ **Mantenimiento Predictivo:** Es mantenimiento basado en condiciones, ya busca que el mantenimiento se realice en el momento que verdaderamente se requiere. Para acertar en lo anterior es necesario hacer mediciones

continuas de algunas variables, o conocer la vida útil promedio de algunos componentes, a fin de reemplazarlos justo antes de que fallen.

d) Requerimiento de recurso humano¹⁷

Además del personal actual de la Asamblea General Universitaria y Asambleístas capacitados con el Uso del Sistema informático desarrollado, será necesaria la incorporación de un nuevo elemento con conocimientos técnicos que de soporte al Sistema en lo relaciona a la administrar la base de datos, mantenimiento de hardware y software, que además sea capaz de modificar el sistema desarrollado a conveniencia de la unidad que lo adquiera.

Por lo antes expuesto, las características que debe cumplir el **Administrador del sistema** del Sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador, se muestran en la descripción de la Tabla 7.

¹⁷ La persona encargada de la Administración del sistema informáticos, puede solicitarse a la dirección de Tecnologías de la UES, en vista que ya es un recurso con los que cuenta la Institución de la cual depende la Unidad Estudiada.

Tabla 7. Descripción del Administrador del sistema
Fuente: Elaboración propia.

Nombre del Puesto	Administrador del sistema
Descripción general	
<p>Persona encargada en las modificaciones necesarias en Los sistemas informáticos a cargo de la Asamblea General Universitaria, así como, de los cambios de los equipos informáticos a cargo de esta última a efectos de garantizar el buen funcionamiento de los Sistemas empleados por la unidad.</p>	
Responsabilidades y Deberes	
<ol style="list-style-type: none"> 1. Brindar soporte Técnico a la Unidad. 2. Realizar los cambios solicitados por la Unidad, a los sistemas informáticos que se encuentran a su disposición. 3. Realizar actividades inherentes al puesto. 4. Garantizar la custodia de información digital en medios magnéticos. 5. Cumplir con las políticas de privacidad de la información manejadas por la Asamblea General Universitaria. 6. Colaborar en la sugerencia de adquisición de equipo informático. 7. Elaborar informes de fallas sobre los Sistemas informáticos a cargo. 8. Apoyar a la unidad en lo solicitado por las autoridades de las mismas. 9. Velar por el buen uso del equipo informático a cargo de la Unidad. 	
Formación Académica	
<p>Técnico o Graduado en Licenciatura en Ciencias de la Computación, Ingeniero de sistemas Informáticos</p>	
Conocimientos Técnicos y Habilidades	
<ul style="list-style-type: none"> ✓ Framework: Laravel, Symfony u otros para PHP ✓ Lenguaje de programación: PHP V5 ✓ Base de datos: MySQL- MariaDB (nivel intermedio) ✓ Servidor: Apache ✓ Conocimientos avanzados en Excel ✓ Conocimiento sobre mantenimiento de Equipos Informáticos ✓ Persona proactiva, honesta 	
Experiencia.	
<p>Mínima de 2 años</p>	

e) Requerimientos de Seguridad

Los requerimientos de seguridad que se contemplan como políticas para garantizar la integridad y confidencialidad de la información siendo los más relevantes para el desarrollo del proyecto el siguiente:

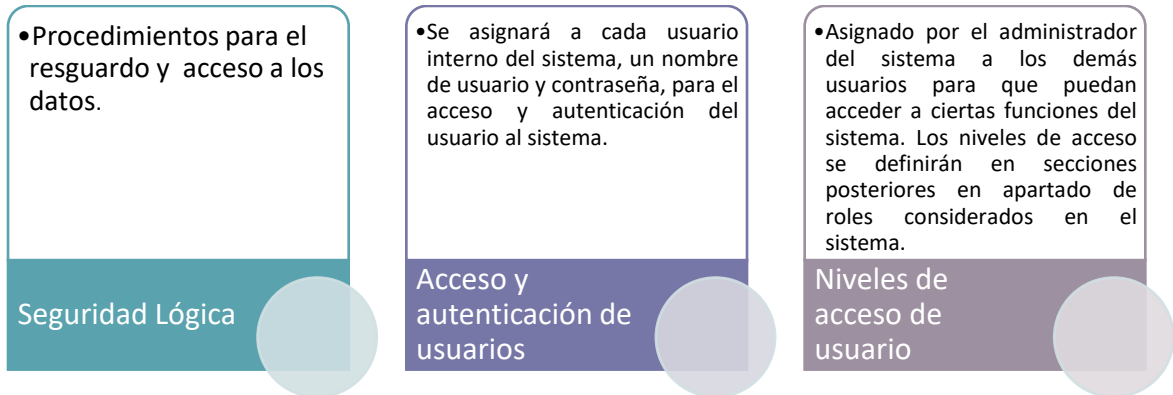


Figura 11. Requerimientos Operativos de Seguridad.

Fuente: Elaboración propia, (BELTRAN SERRANO , SANCHEZ CASTRO, LANDAVERDE GARCIA, BOLAÑOS CANJURA, & MIRANDA RAMIREZ, 2017)

1.2.6. Diagrama de enfoque de la solución propuesta

“Sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de el salvador

OBJETIVO: Desarrollar un Sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador

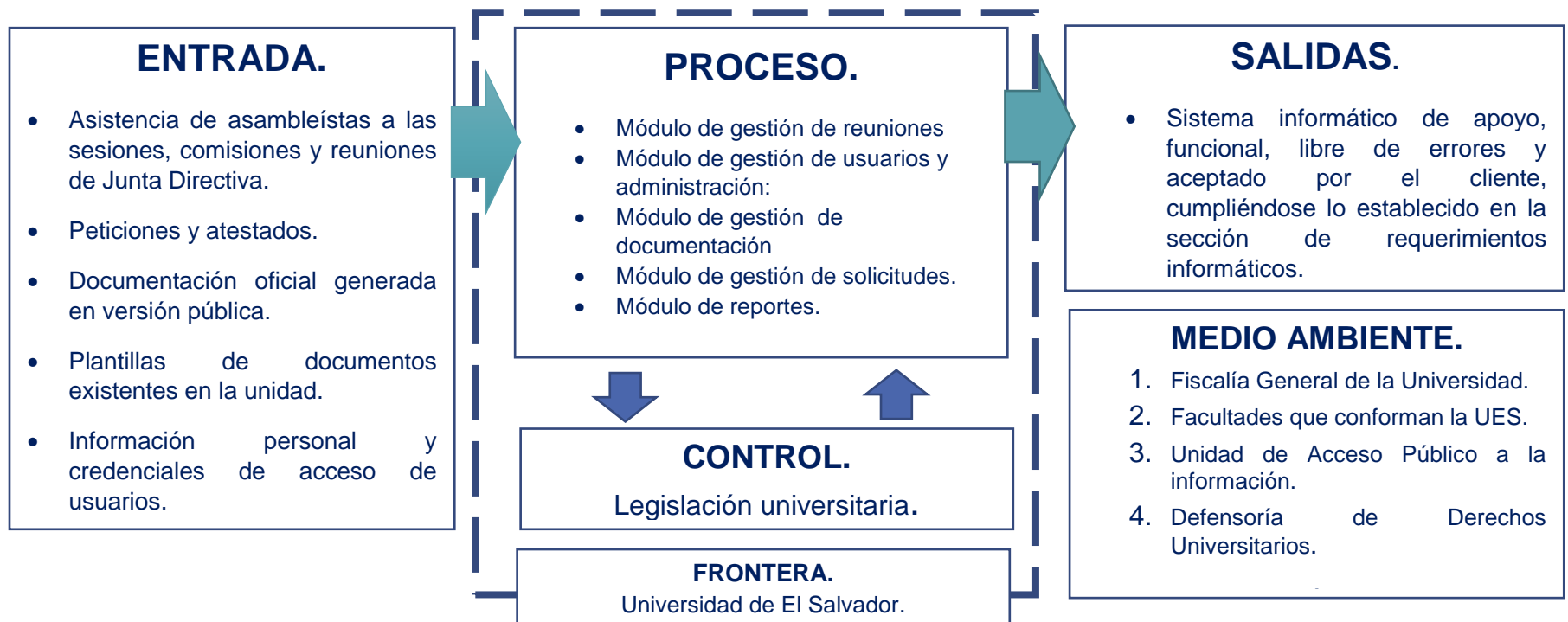


Figura 12. Enfoque de Sistema
Fuente: Elaboración Propia

1.2.7. Casos de uso¹⁸

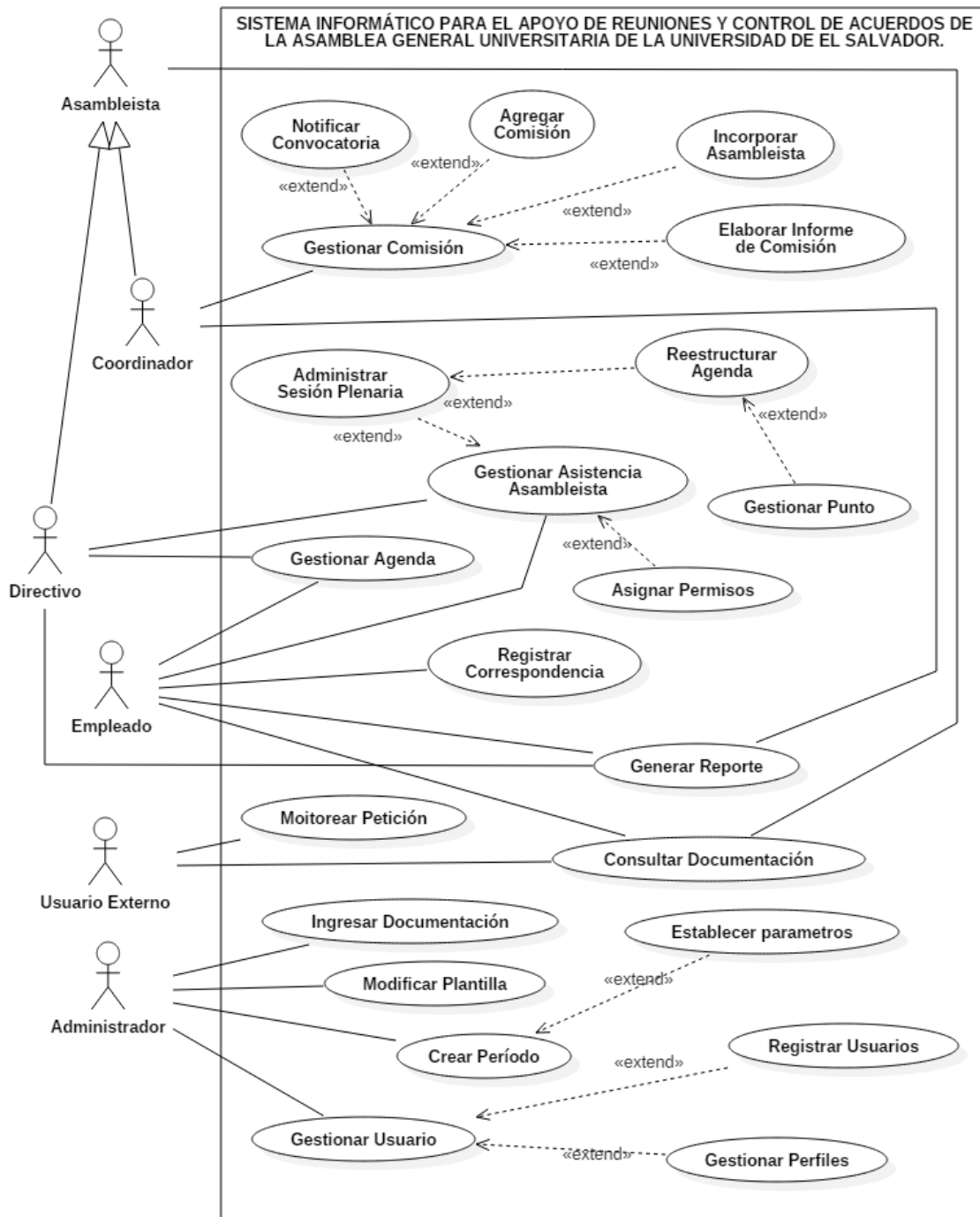


Figura 13. Diagrama de Caso de Uso General
Fuente: Elaboración Propia, StartUML

¹⁸ A partir de aquí, se considera la información obtenida por el cuestionario de análisis y diseño del anexo e)

1.2.7.1. Lista Actor Objetivo

Tabla 8. Lista Actor-Objetivo
Fuente: Elaboración propia

ACTOR	OBJETIVO
ASAMBLEÍSTA DIRECTIVO	Consultar Documentación Gestionar Agenda Gestionar Asistencia Asambleísta Asignar Permisos Administrar Sesión Plenaria Reestructurar Agenda Gestionar Punto Generar Reporte
COORDINADOR	Gestionar Comisión Notificar Convocatoria Agregar Comisión Incorporar Asambleísta Elaborar Informe de Comisión Generar Reporte
EMPLEADO	Gestionar Agenda Gestionar Asistencia Asambleísta Asignar Permisos Administrar Sesión Plenaria Reestructurar Agenda Gestionar Punto Generar Reportes. Consultar Documentación. Registrar Correspondencia.
ADMINISTRADOR	Gestionar Usuarios Gestionar Perfiles. Registrar Usuarios Crear Periodo Establecer parámetros Modificar Plantilla. Ingresar Documentación
USUARIO EXTERNO	Monitorear Petición Consultar Documentación

1.2.7.2. Especificación de Casos de Usos

El formato utilizado para la especificación de caso de uso, se establece en la tabla No.9.

Tabla 9. Formato de especificación de Caso de Uso

Fuente: (ALAVA, 2015)

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Nombre del caso de uso
ACTOR	Nombre del actor que realiza el caso de uso
DESCRIPCIÓN	Descripción del funcionamiento del caso de uso
FLUJO BÁSICO	Descripción del flujo normal e ininterrumpido descrita de manera secuencial. Ej. <ol style="list-style-type: none">1. Descripción del paso 12. Descripción del paso 23. Etc.
FLUJO ALTERNATIVO	Descripción del flujo Alternativo que puede tener cada caso de uso descrita de manera secuencial. Ej. <ol style="list-style-type: none">1. Descripción del paso 12. Descripción del paso 23. Etc.
PRE-CONDICIÓN	Condición o condiciones que se deben de cumplir para iniciar el caso de uso
POST-CONDICIÓN	Condición o condiciones que se cumplen tras la realización correcta del caso de uso

- **Caso de Uso: Monitorear Petición**

Tabla 10. Especificación CU - Monitorear Petición
Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Monitorear Petición
ACTOR	Usuario Externo
DESCRIPCIÓN	<p>El usuario externo que haya realizado una solicitud previamente podrá ver el estado de dicha solicitud entrando a la página web pública de la aplicación entrando en la opción de estado de solicitudes y realizando una búsqueda de su solicitud mediante el buscador o el listado previamente cargado.</p> <p>Las peticiones se desplegarán en un listado en el cual se mostrar un nombre de petición el nombre del solicitante y una pequeña descripción, además se mostrará una barra de colores para cada petición la cual indicara el tiempo que ha transcurrido desde su entrada al proceso de revisión.</p>
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. Entrar al sitio web de dominio público de la aplicación SIARCA. 2. Ir al menú de estado de peticiones. 3. Buscar petición. 4. Revisar estado
FLUJO ALTERNATIVO	<p>Flujo en caso de errores.</p> <ol style="list-style-type: none"> 1. La aplicación proporcionara un mensaje de error y proporcionara un enlace a la página principal.
PRE-CONDICIÓN	Haber realizado una solicitud en la recepción de solicitudes de la asamblea general universitaria.
POST-CONDICIÓN	Se desplegará la información de la petición a revisar

i. DSS: Flujo normal de Éxito

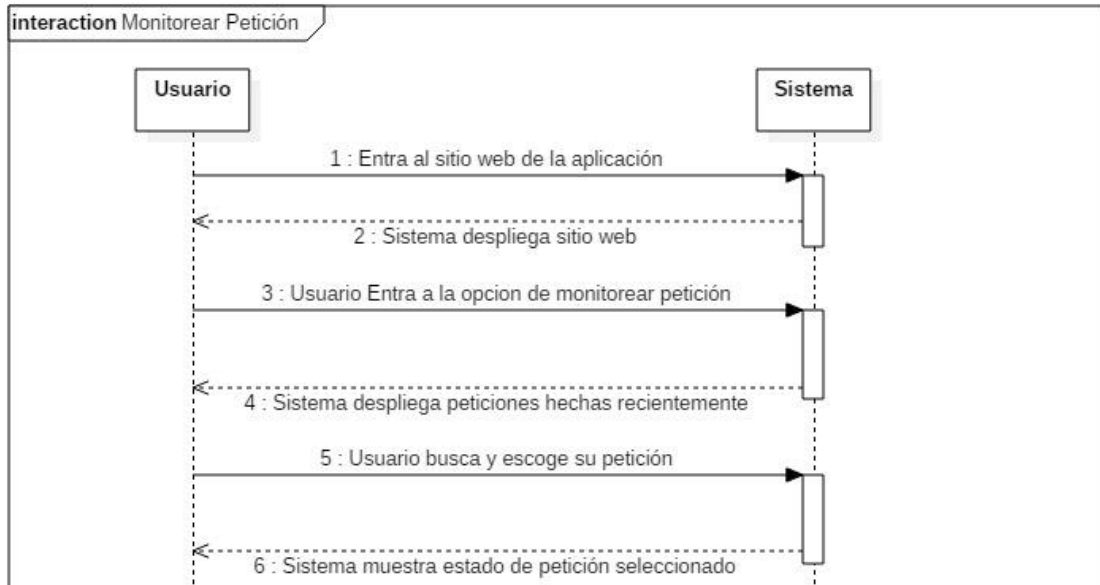


Figura 14 DSS- Monitorear Petición (Flujo Básico)
Fuente: Elaboración Propia, starUML

ii. DSS: Flujo normal de Alternativo

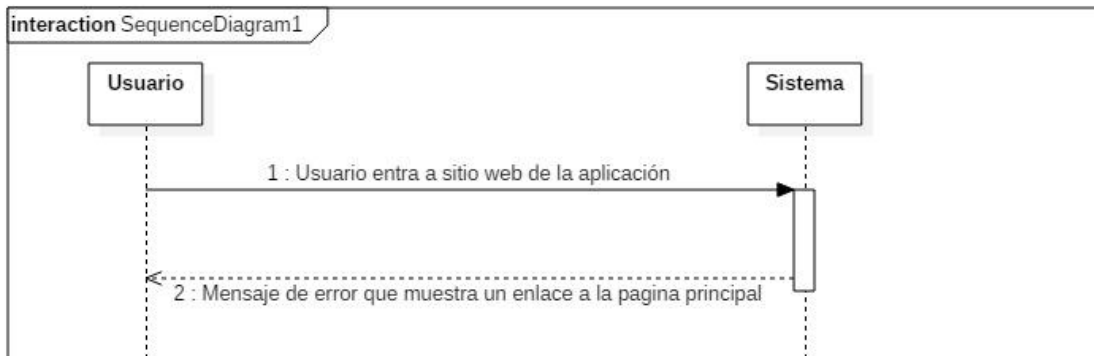


Figura 15 DSS- Monitorear Petición (Flujo Alternativo)
Fuente: Elaboración Propia, starUML

- **Caso de Uso: Agregar Comisión**

Tabla 11. Especificación CU - Agregar Comisión
Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Agregar Comisión
ACTOR	Asambleísta, Coordinador de Comisión
DESCRIPCIÓN	El asambleísta accederá al menú asociado a las comisiones, en el cual encontrará la opción de crear una comisión, después de haber hecho clic a esta, el sistema le permitirá especificar al usuario el nombre de la comisión y por medio de un botón de tipo switch On/off determinar si la comisión estará activa o no.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El asambleísta accede a la opción de crear una comisión al sistema. 2. El sistema muestra un formulario solicitando los siguientes datos de la nueva comisión: 3. El usuario ingresa los datos solicitados. 4. El sistema verifica que los datos estén completos 5. El sistema verifica que no exista la comisión en el sistema. 6. El sistema registra la nueva comisión. 7. El sistema muestra mensaje de éxito
FLUJO ALTERNATIVO	<p>4.a. El sistema determina que los datos están incompletos.</p> <ol style="list-style-type: none"> 1. El sistema informa que campos están incompletos y solicita su ingreso. 2. Se vuelve al paso 3 del flujo normal de éxito. <p>5.a. El sistema determina que la comisión ya existe, solicita cambio</p>
PRE-CONDICIÓN	Haber ingresado como usuario registrado en el sistema y poseer los permisos necesarios.
POST-CONDICIÓN	Se agrega una comisión exitosamente

i. DSS: Flujo normal de Éxito

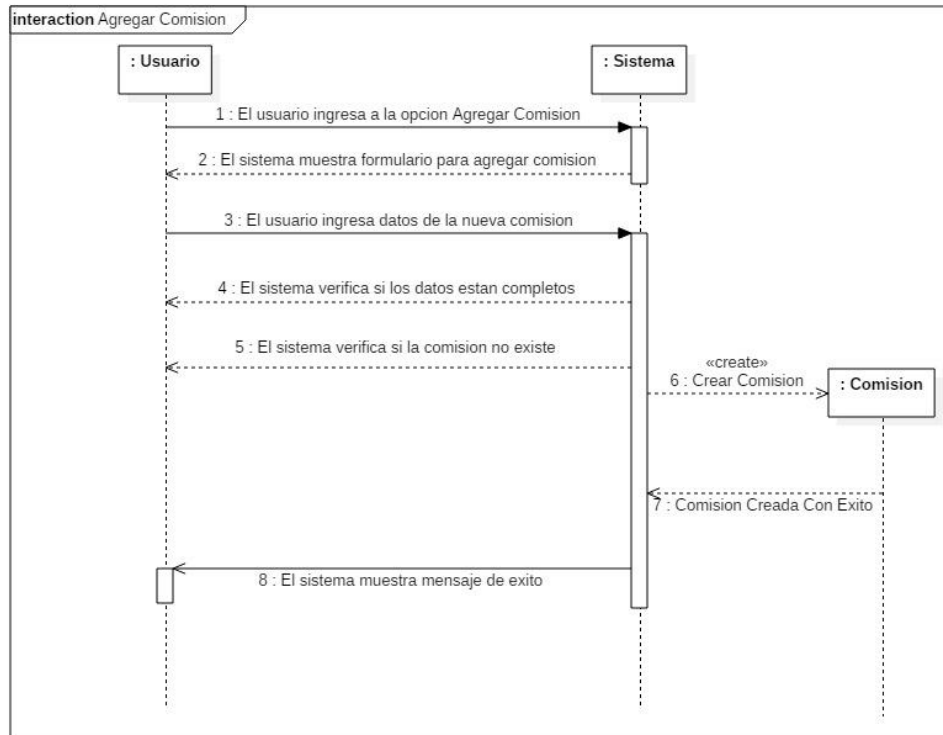


Figura 16. DSS- Agregar Comisi3n (Flujo B3sico)
Fuente: Elaboraci3n Propia, StarUML

ii. DSS: Flujo Alternativo 4.a

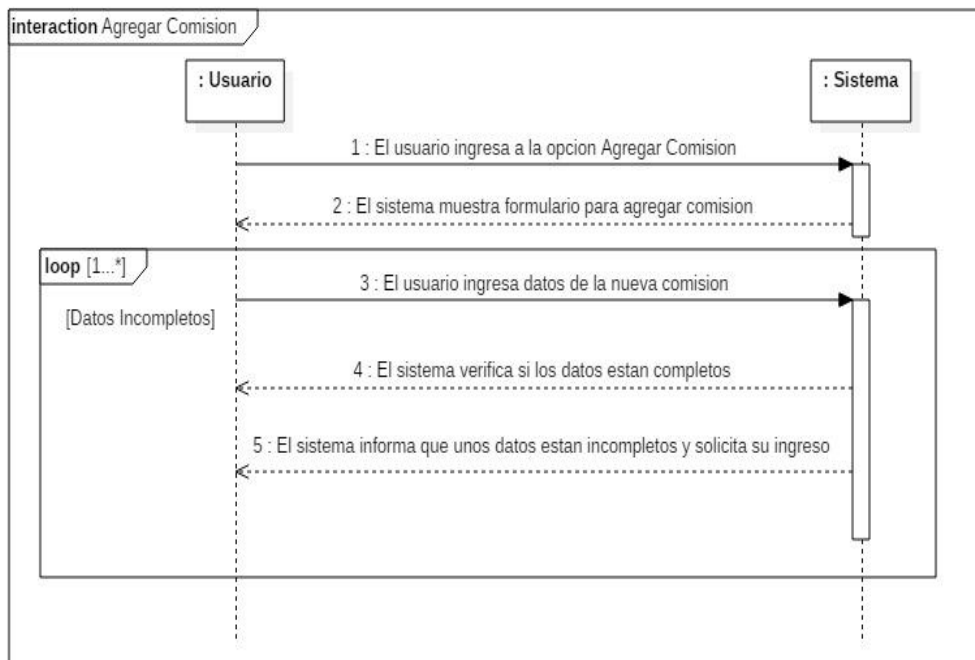


Figura 17. DSS- Agregar Comisi3n (Flujo Alternativo 4.a)
Fuente: Elaboraci3n Propia, StarUML

iii. DSS: Flujo Alternativo 5.a

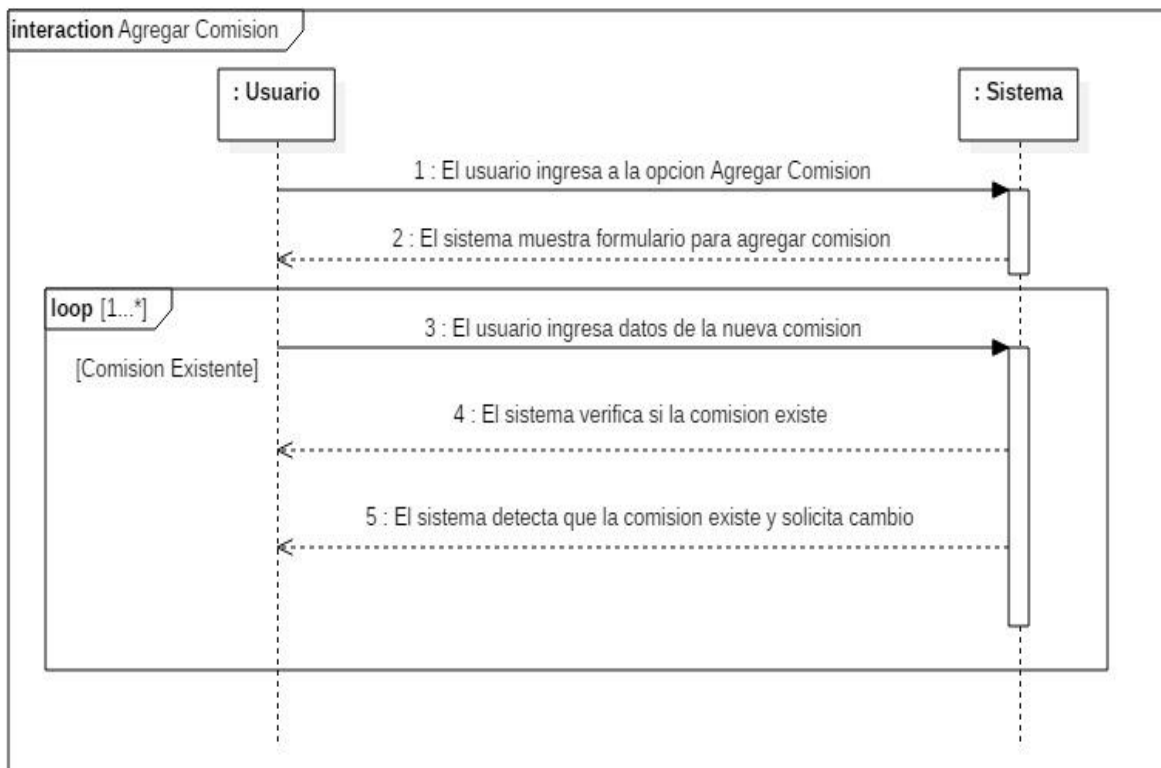


Figura 18. DSS- Agregar Comisión (Flujo Alternativo 5.a)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Incorporar Asambleísta**

Tabla 12. Especificación CU - Incorporar Asambleísta
Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Incorporar Asambleísta
ACTOR	Coordinador de Comisión
DESCRIPCIÓN	El asambleísta agrega a una persona para que esta forme parte de una comisión y sea capaz de participar en todas sus actividades
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El asambleísta accede a la opción de agregar asambleísta. 2. El sistema presenta un formulario solicitando la comisión y las personas a agregar a dicha comisión. 3. El usuario selecciona la comisión 4. El usuario agrega los asambleístas a la comisión seleccionada. 5. El usuario solicita registrar los cambios 6. El sistema valida los datos 7. El sistema registra los cambios 8. El sistema notifica que se realizó la operación con éxito.
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 6.a. El sistema determina que los datos están incompletos o erróneos. <ol style="list-style-type: none"> 1. El sistema informa que no se pudo efectuar los cambios y solicita que se ingresen nuevamente 2. Se vuelve al paso 2 del flujo normal de éxito.
PRE-CONDICIÓN	Haber ingresado como usuario registrado en el sistema y poseer los permisos necesarios.
POST-CONDICIÓN	Se agrega una persona a una comisión exitosamente

i. DSS Flujo Normal de Éxito

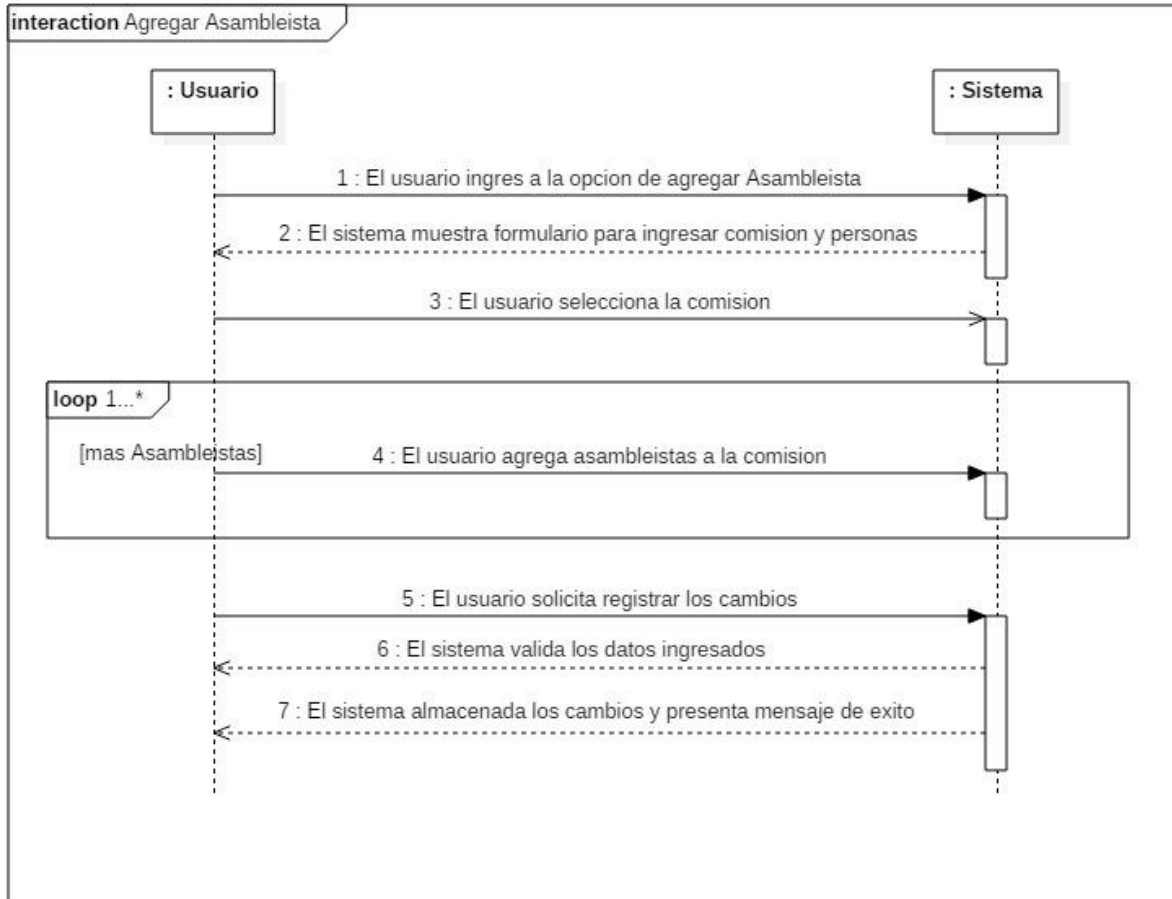


Figura 19. DSS- Agregar Asambleísta (Flujo Básico)
Fuente: Elaboración Propia, StarUML

ii. DSS Flujo Alternativo 6.a

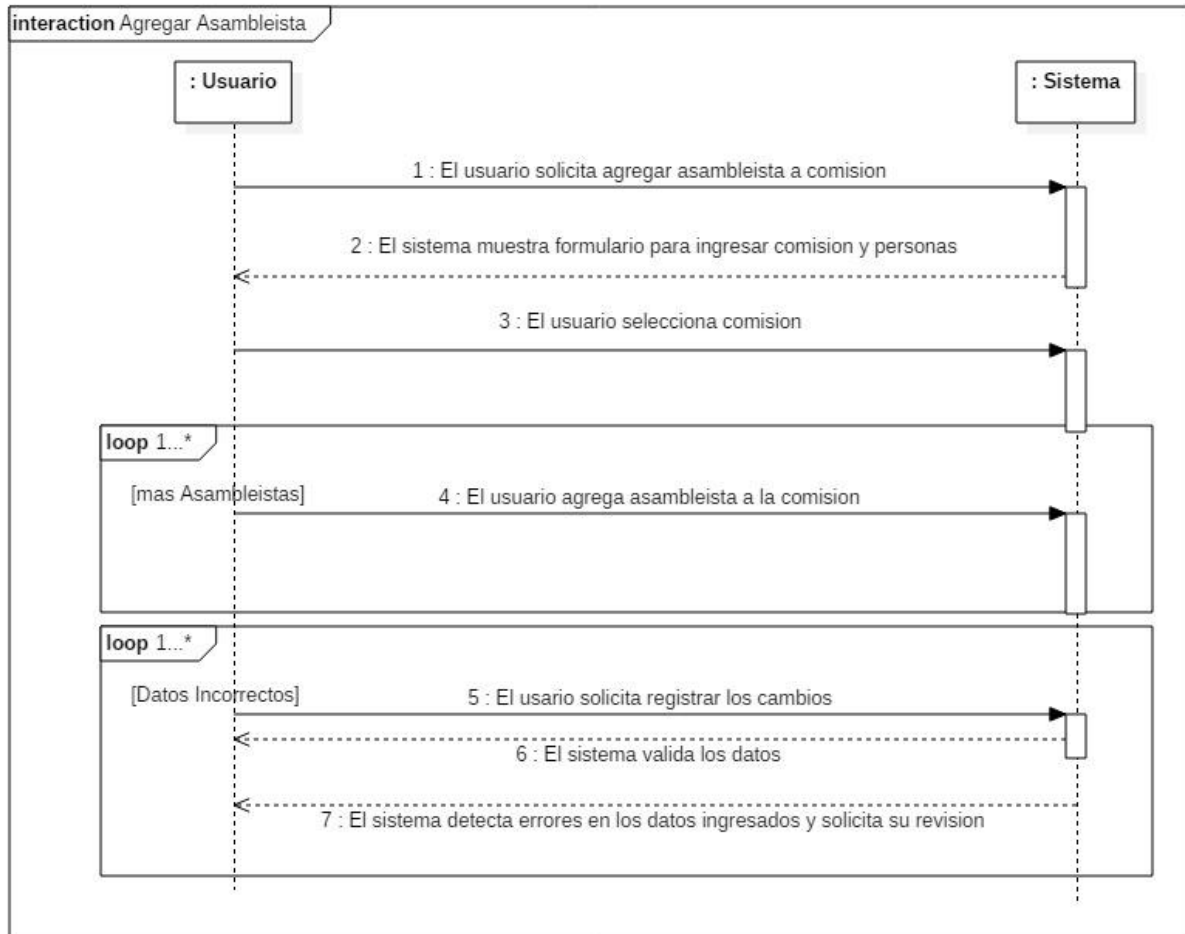


Figura 20. DSS- Agregar Asambleísta (Flujo Alternativo 6.a)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Notificar Convocatoria**

Tabla 13. Especificación CU - Notificar Convocatoria
Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Notificar Convocatoria
ACTOR	Asambleísta, Coordinador de Comisión
DESCRIPCIÓN	El asambleísta notifica por medio de un correo electrónico a todas las personas que conforman una comisión de la próxima sesión detallando la fecha y la hora.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El asambleísta accede a la opción de notificar convocatoria. 2. El sistema presenta una pantalla en la que el usuario debe de ingresar el contenido de la convocatoria y la comisión, la cual el usuario desea notificar de la próxima reunión. 3. El usuario ingresa los datos, solicita guardar los cambios y enviar la notificación. 4. El sistema valida los datos ingresados. 5. El sistema envía la notificación a todos los asambleístas que forman parte de esa comisión. 6. El sistema informa que la notificación se envió de manera exitosa.
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 4.a. El sistema determina que los datos están incompletos o erróneos. <ol style="list-style-type: none"> 1. El sistema informa que campos están incompletos y solicita su ingreso. 2. Se vuelve al paso 3 del flujo normal de éxito.
PRE-CONDICIÓN	Haber ingresado como usuario registrado en el sistema y poseer los permisos necesarios.
POST-CONDICIÓN	Se notifica a los asambleístas que conforman la comisión de la próxima sesión exitosamente

i. DSS Flujo Normal de Éxito

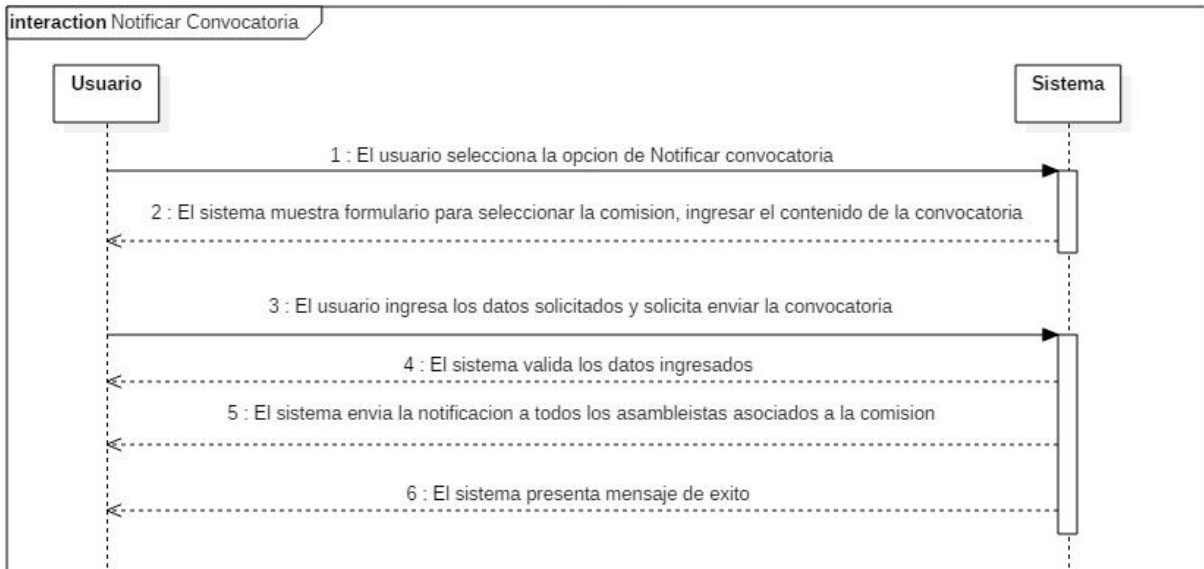


Figura 21. DSS- Notificar Convocatoria (Flujo Básico)
Fuente: Elaboración Propia, StarUML

ii. DSS Flujo Alternativo 4.a

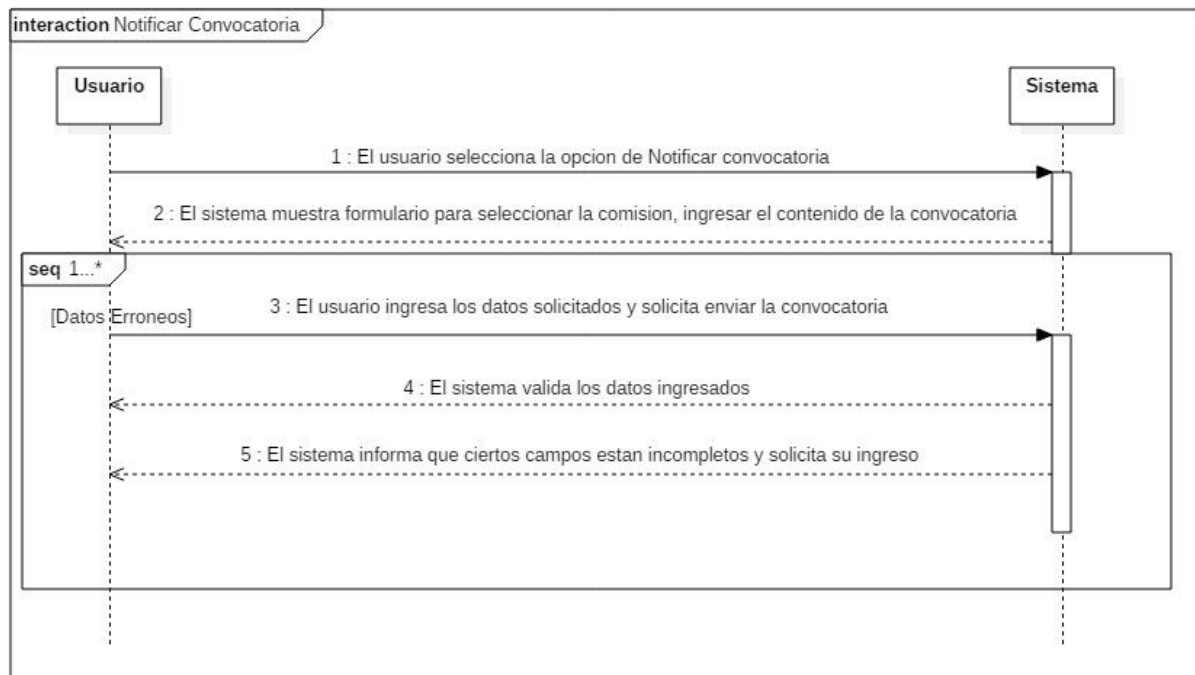


Figura 22. DSS- Notificar Convocatoria (Flujo Alternativo 4.a)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Gestionar Comisión**

Tabla 14. Especificación CU - Gestionar Comisión
Fuente: Elaboración Propia

	DESCRIPCIÓN
CASO DE USO	Gestionar Comisión
ACTOR	Coordinador de Comisión
DESCRIPCIÓN	El usuario por medio de esta opción será capaz de llevar a cabo la administración de la comisión, ya sea integrar o retirar asambleístas a esta o registrar el trabajo que en esta se realiza.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El usuario hace clic sobre la opción Comisión 2. El sistema le presenta una pantalla con las posibles opciones a realizar dentro del sistema relacionado con las Comisiones. 3. El usuario selecciona una de las posibles opciones 4. El sistema le presenta la interfaz asociada con la opción seleccionada
FLUJO ALTERNATIVO	N/A
PRE-CONDICIÓN	Haber ingresado como usuario registrado en el sistema y poseer los permisos necesarios.
POST-CONDICIÓN	El sistema le presenta la interfaz de la opción seleccionada

i. DSS Flujo Normal de Éxito

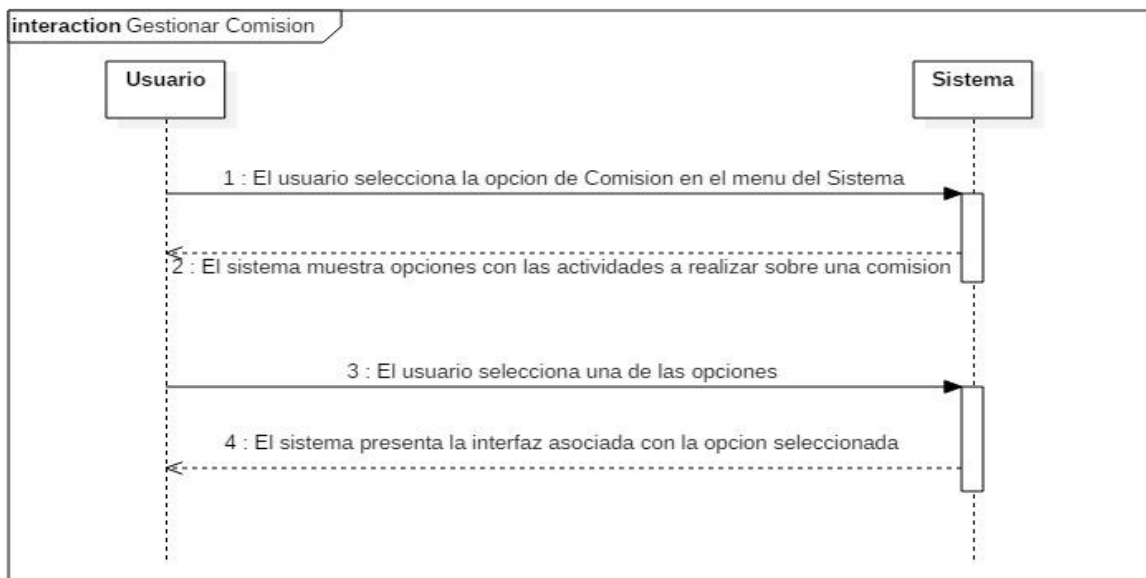


Figura 23. DSS- Gestionar Comisión (Flujo Básico)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Registrar Peticiones**

Tabla 15. Especificación CU – Registrar Peticiones
Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Registrar Peticiones
ACTOR	Empleado
DESCRIPCIÓN	<p>Este caso de uso comienza cuando un usuario externo llega a el registro de peticiones de la asamblea general universitaria para dejar una petición que quiere que sea resulta por la AGU. La secretaria recibe los documentos y registra la petición en el sistema ingresando los siguientes datos:</p> <ol style="list-style-type: none"> 1. Remitente 2. Descripción 3. Documento adjunto <p>Al ser insertada la petición en el sistema esta entrara en cola y se mostrara en el menú de peticiones en espera de ser procesada como un tema para discusión.</p>
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. Ingresar remitente 2. Ingresar Descripción 3. Ingresar Documento adjunto 4. Ingresar
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 1. Flujo en caso de errores. Las vistas de registro de ingreso de datos tendrán todos los filtros necesarios para que los datos ingresados sean los más confiables posibles. 2. Flujo en caso de cancelación. <ol style="list-style-type: none"> 1. El usuario cancela el ingreso de petición. 2. Se muestra un mensaje preguntado si está seguro que desea cancelar el ingreso de datos. 3. El usuario acepta el mensaje 4. El usuario es regresado a la pantalla previa al ingreso de peticiones o al menú inicial.
PRE-CONDICIÓN	<ol style="list-style-type: none"> 1. Haber entrado con un usuario registrado en la aplicación. 2. Entrar al menú de registro de peticiones
POST-CONDICIÓN	<ol style="list-style-type: none"> 1. Mensaje de confirmación que los datos fueron registrados correctamente. 2. Petición puede ser visualizado en el menú de peticiones

- **DSS Flujo Normal de Éxito**

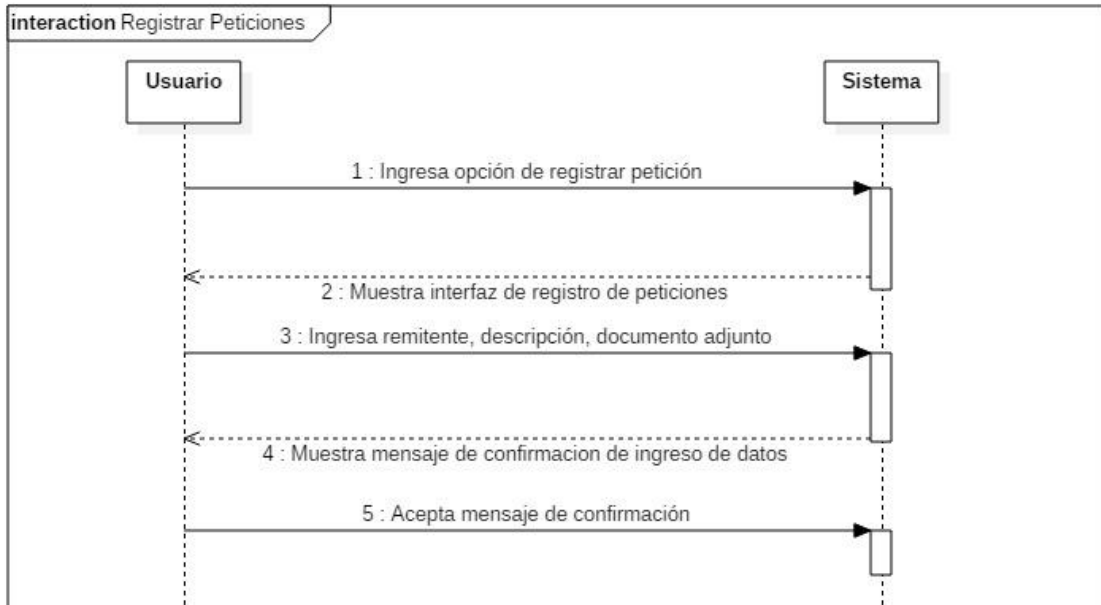


Figura 24: DSS- Registrar Peticiones (Flujo Básico)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Gestionar Perfiles**

Tabla 16. Especificación CU - Gestionar Perfiles
Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Gestionar perfiles
ACTOR	Administrador
DESCRIPCIÓN	Existen diferentes tipos de perfiles que se manejan en el sistema: <ol style="list-style-type: none"> 1. Asambleísta 2. Coordinador comisión 3. Directivo 4. Empleado 5. Administrador <p>Los cuáles serán administrados desde este caso de uso, el cual consiste en agregar o denegar permisos a las diferentes funcionalidades del sistema, apegándonos siempre a la lógica del negocio que se maneja.</p>
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. Ingresar a la opción de gestionar perfiles 2. Escoger un usuario de la lista o buscarlo 3. Hacer clic en el botón permisos 4. Agregar o denegar permisos según perfil de usuario
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 3.a. Flujo en caso de cancelación. <ol style="list-style-type: none"> 3.1. El usuario cancela el ingreso de petición. 3.2. Se muestra un mensaje preguntado si está seguro que desea cancelar el ingreso de datos. 3.3. El usuario acepta el mensaje 3.4. El usuario es regresado a la pantalla previa al ingreso de peticiones o al menú inicial.
PRE-CONDICIÓN	<ol style="list-style-type: none"> 1. Haber entrado con un usuario registrado en la aplicación. 2. Entrar al menú de Administración de perfiles
POST-CONDICIÓN	<ol style="list-style-type: none"> 1. Mensaje de confirmación que los datos fueron registrados correctamente. 2. Perfiles puede ser visualizado en el menú de Perfiles

i. DSS Flujo Normal de Éxito

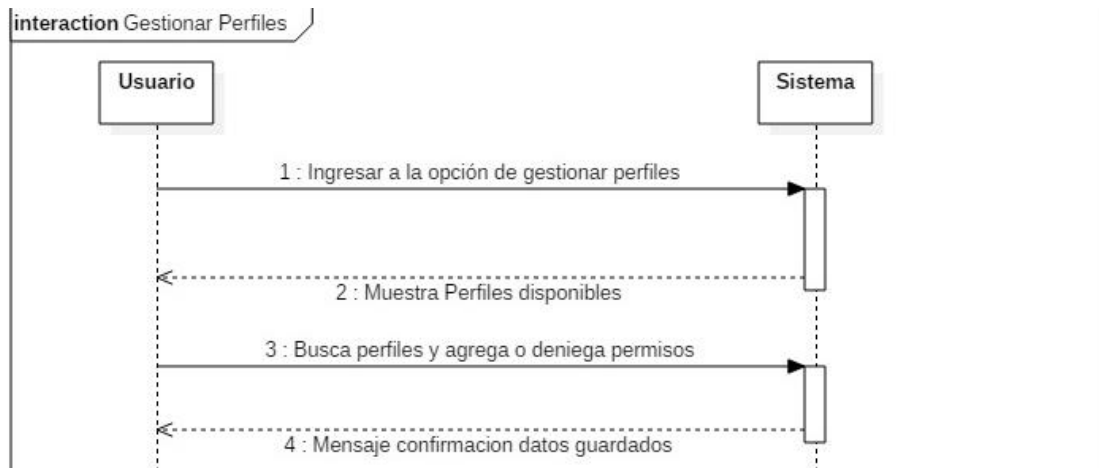


Figura 25 DSS: Gestionar Perfiles (Flujo Básico)
Fuente: Elaboración Propia, StarUML

ii. DSS Flujo Alternativo 3.a

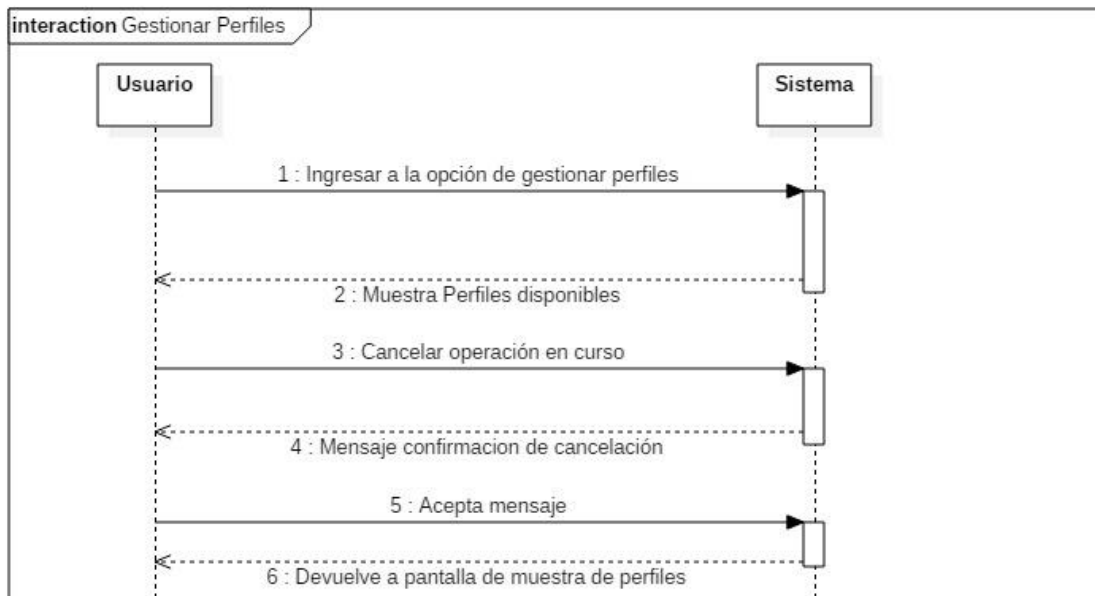


Figura 26 DSS: Gestionar Perfiles (Flujo Alternativo)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Gestionar Usuario**

Tabla 17. Especificación CU - Gestionar Usuario
Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Gestionar Usuario
ACTOR	Empleado, Administrador
DESCRIPCIÓN	<p>Este caso de uso se refiere a el manejo de los usuarios que accederán al sistema, cada usuario tendrá asignado un perfil de usuario el cual definirá las funcionalidades a las cuales podrá acceder dentro del sistema, se crearan usuarios para cada miembro asambleísta y para el personal administrativo.</p> <p>Podremos tener la opción de cargar archivos Excel con un formato preestablecido para poder ingresar usuarios de manera rápida y manual.</p>
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. Ingresar a la opción gestionar usuario 2. Ingresar nombre usuario, contraseña, correo y perfil de usuario. 3. Confirmar ingreso.
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 1.a. Ingresar a la opción gestionar usuario <ol style="list-style-type: none"> 1. Escoger opción ingreso mediante medios externos 2. Escoger documento Excel que contendrán los usuarios en el formato preestablecido. 3. Confirmar ingreso.
PRE-CONDICIÓN	<ol style="list-style-type: none"> 1. Haber ingresado como usuario registrado en el sistema y poseer los permisos necesarios.
POST-CONDICIÓN	<ol style="list-style-type: none"> 1. Mensaje de confirmación que los cambios fueron efectuado exitosamente

i. DSS Flujo Normal de Éxito

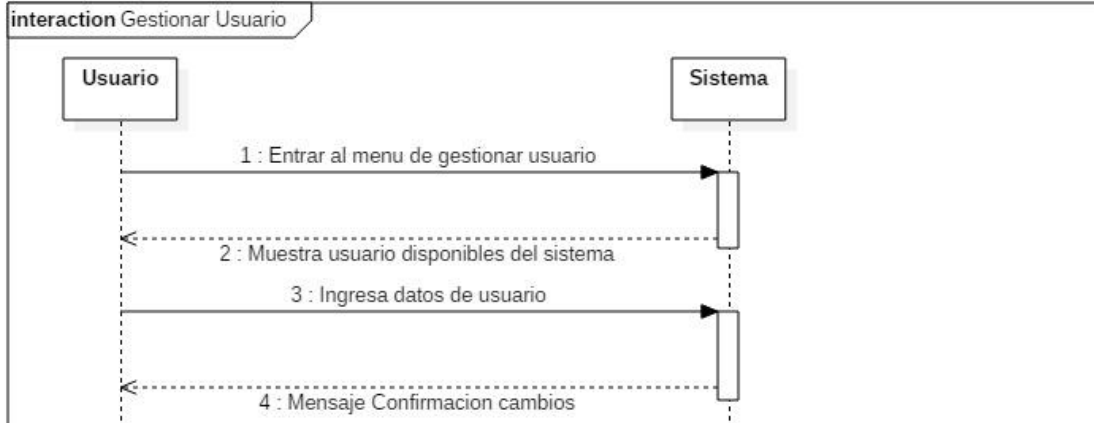


Figura 27 DSS: Gestionar Usuario (Flujo Básico)
Fuente: Elaboración Propia, StarUML

ii. DSS Flujo Alternativo 1.a

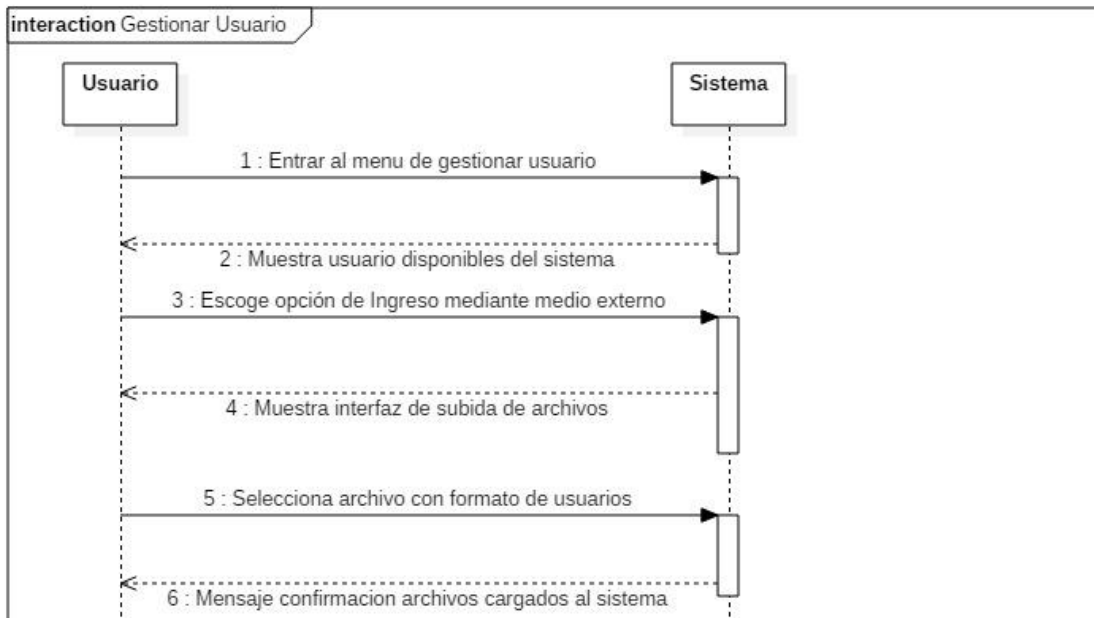


Figura 28 DSS: Gestionar Usuario (Flujo Básico)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Registrar Asambleísta**

Tabla 18. Especificación CU - Registrar Asambleísta
Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Registrar Asambleísta
ACTOR	Empleado, Administrador
DESCRIPCIÓN	Este caso de uso permite agregar un nuevo asambleísta a una nueva sesión plenaria, la cual deberá de ser iniciada previamente.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. Entrar a la primera pantalla de nueva sesión plenaria. 2. Ingresar nombre asambleísta. 3. Agregar asambleísta. 4. Escoger facultad para visualizar asambleístas previos
FLUJO ALTERNATIVO	<p>3.a. ingresar asambleísta ya inscrito.</p> <ol style="list-style-type: none"> 1. Usuario ingresa nombre de usuario ya inscrito. 2. se muestra un mensaje que el nombre ya ha sido ingresado. 3. Usuario acepta el mensaje y es devuelto al menú de ingreso de asambleístas
PRE-CONDICIÓN	<ol style="list-style-type: none"> 1. Haber ingresado como usuario registrado en el sistema y poseer los permisos necesarios. 2. Haber iniciado una nueva sesión plenaria
POST-CONDICIÓN	<ol style="list-style-type: none"> 1. Usuario agregado al listado correspondiente a su facultad.

i. DSS Flujo Normal de Éxito

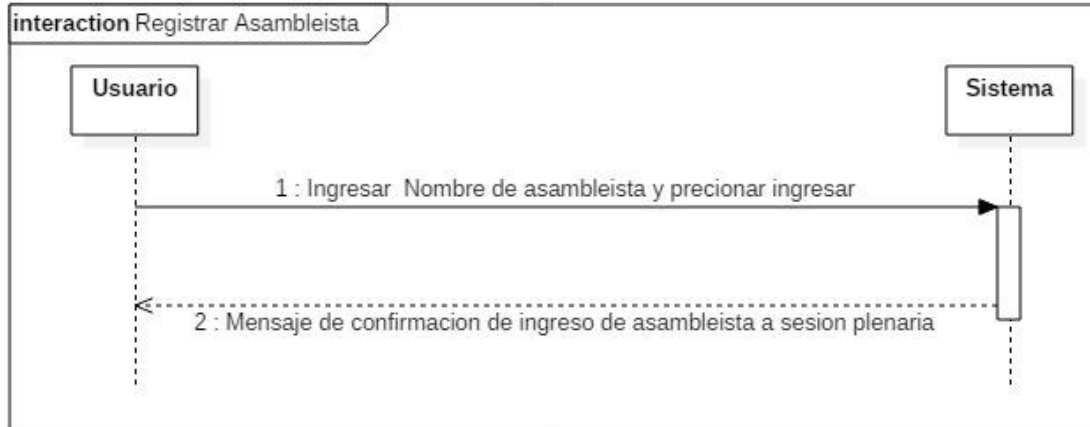


Figura 29 DSS: Registrar Asambleísta (Flujo Básico)
Fuente: Elaboración propia, StarUML

ii. DSS Flujo Alternativo 3.a

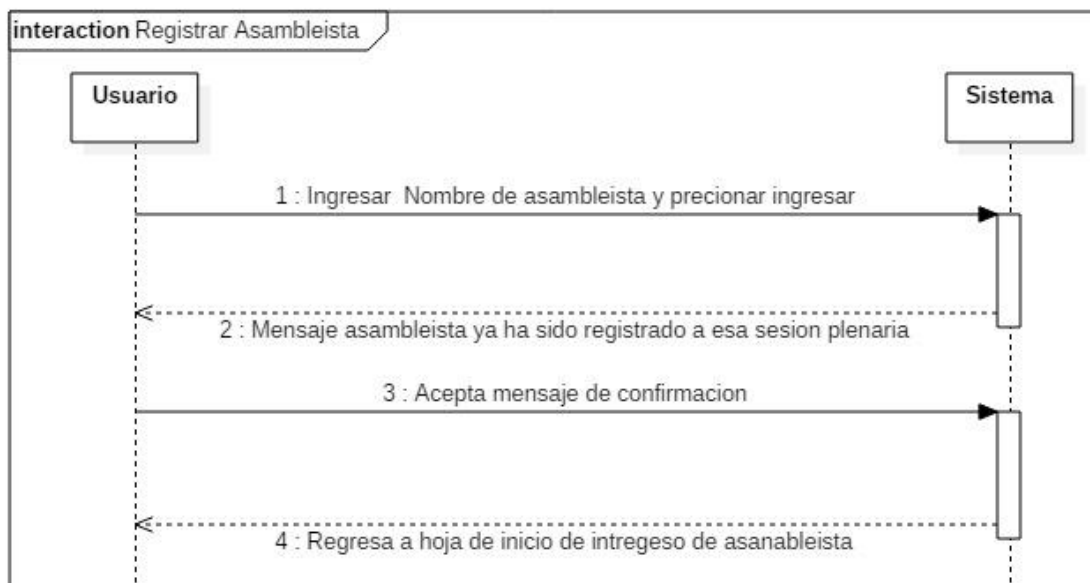


Figura 30 DSS: Registrar Asambleísta (Flujo Alternativo)
Fuente: Elaboración Propia

- **Caso de Uso: Gestionar Agenda**

Tabla 19. Especificación CU - Gestionar Agenda
Fuente: Elaboración propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Gestionar Agenda
ACTOR	Directivo, Empleado (rol: Profesional Universitario)
DESCRIPCIÓN	Este caso de uso permitirá la elaboración de agendas propuestas a las sesiones plenarias realizados por los asambleístas a partir de la correspondencia previamente ingresada al sistema y listada por ítems según fecha de ingreso y criterio de importancia. Se permitirá, además, la especificación de día, hora y lugar de realización.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El sistema muestra un formulario para especificaciones generales de la reunión próxima, además de mostrar el listado de los puntos pendientes a ser agendados, ordenados de forma ascendentes por fecha y atestados respectivos. 2. El actor llena los campos disponibles. 3. El actor selecciona los ítems a ser agendados en la próxima sesión plenaria, indicando el tipo de punto según clasificación de los mismos y guarda los puntos establecidos. 4. El sistema muestra un mensaje para confirmar las generales ingresadas y el total de puntos asignados como agenda propuesta. 5. El actor acepta los cambios realizados.
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 3.a Cuando el punto a agendar no posee atestados. <ol style="list-style-type: none"> 1. El actor Selecciona el punto a agendar sin atestado. 2. El sistema muestra una pestaña de ingreso de archivo. 3. El actor selecciona el icono de ingreso de archivo y carga el archivo. 4. El sistema muestra un mensaje de confirmación de éxito, una vez cargado el documento. 5. Vuelve al flujo básico.
PRE-CONDICIÓN	<ol style="list-style-type: none"> 1. Existencia de un período 2. Haber iniciado sesión 3. Existencia de peticiones en el sistema
POST-CONDICIÓN	<ol style="list-style-type: none"> 1. Mensaje de confirmación para generar Agenda

i. DSS: Flujo normal de Éxito

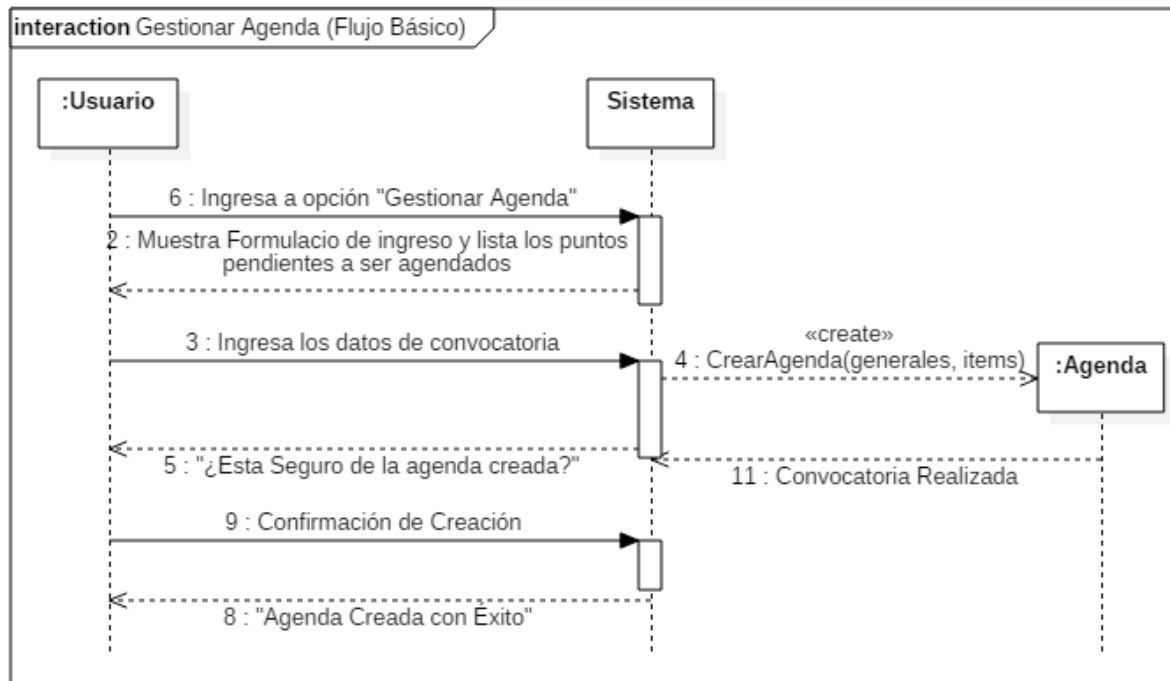


Figura 31. DSS- Gestionar Agenda (Flujo Básico)
Fuente: Elaboración Propia, StarUML

ii. DSS Flujo Alternativo 3.a

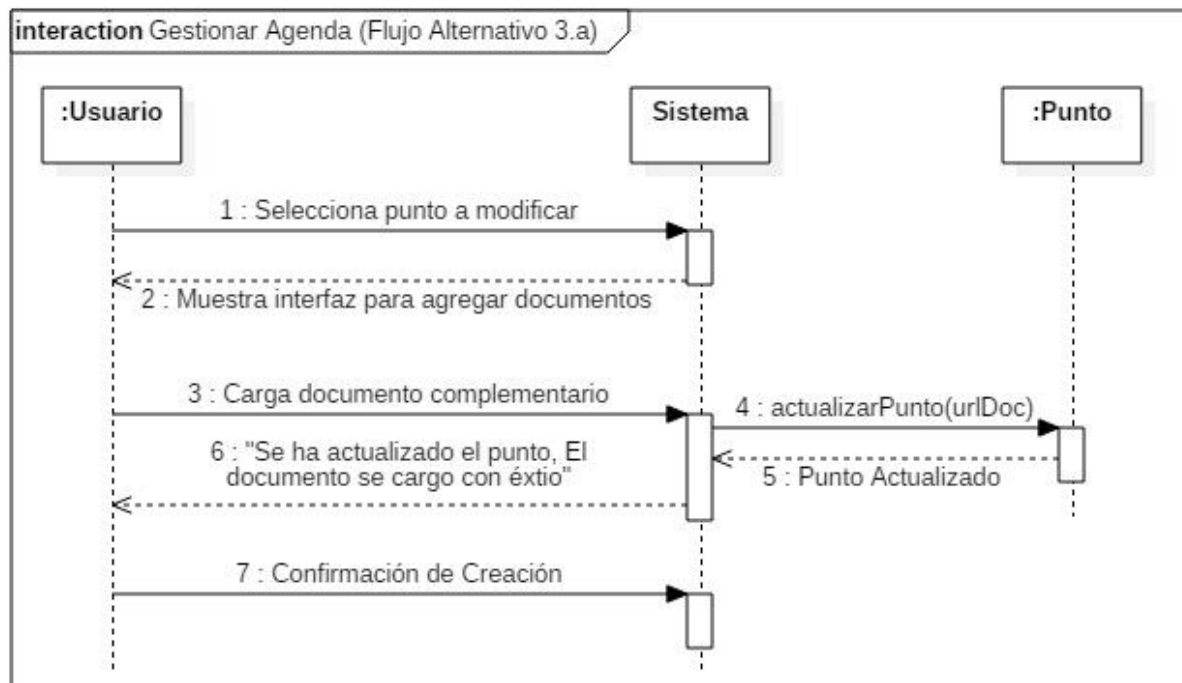


Figura 32. DSS - Gestionar Agenda (Flujo Alternativo 3.a)
Fuente: Elaboración propia, StarUML

- **Caso de Uso: Gestionar Asistencia de asambleístas**

Tabla 20. Especificación CU - Gestionar Asistencia de Asambleísta
Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Gestionar Asistencia de asambleísta
ACTOR	Directivo, Empleado (rol: Profesional Universitario)
DESCRIPCIÓN	Este caso de uso permitirá el ingreso y control de la lista de asistencia a las asambleas. La toma de asistencia será el punto de partida para apertura de sesión y cuantificar el porcentaje de permanencia vinculado al proceso de asignación de dietas.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El sistema muestra un formulario para la búsqueda de asambleístas en el sistema. 2. El actor ingresa el identificador del asambleísta a registrar y selecciona el nombre del asambleísta entrante. 3. El sistema muestra interfaz con lista de los 5 asambleístas más recientes con panel de control para la asistencia por Facultad y detalle de ingreso. 4. El actor inicia caso de uso Administrar sesiones Plenarias¹⁹.
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 3.a. Retiro de Asambleísta. <ol style="list-style-type: none"> 1. El actor Selecciona la facultad a gestionar. 2. El Sistema muestra interfaz para Asignar Permisos. 3, El actor inicia el caso de uso Asignar Permisos. 4. El actor retorna al Flujo básico del caso de uso.
PRE-CONDICIÓN	<ol style="list-style-type: none"> 1. Existencia de un período 2. Haber iniciado sesión 3. Existencia de asambleístas en el sistema
POST-CONDICIÓN	<ol style="list-style-type: none"> 1. Se almacena el registro de las personas asistentes a la sesión. 2. Se habilita Opción de inicio de sesión plenaria.

¹⁹ Esta acción, una vez se establezca el Quorum mínimo para sesionar válidamente de acuerdo al tipo de agenda.

i. **DSS: Flujo normal de Éxito**

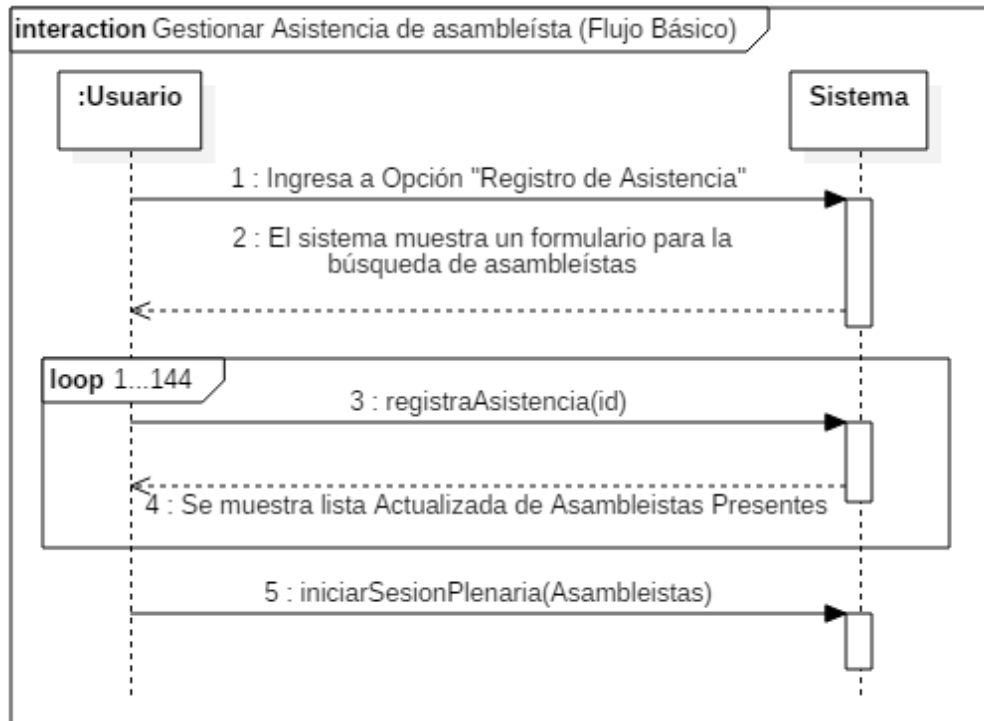


Figura 33. DSS - Gestionar Asistencia de Asambleísta (Flujo Básico)
Fuente: Elaboración Propia, StarUML

ii. **DSS Flujo Alternativo 3.a**

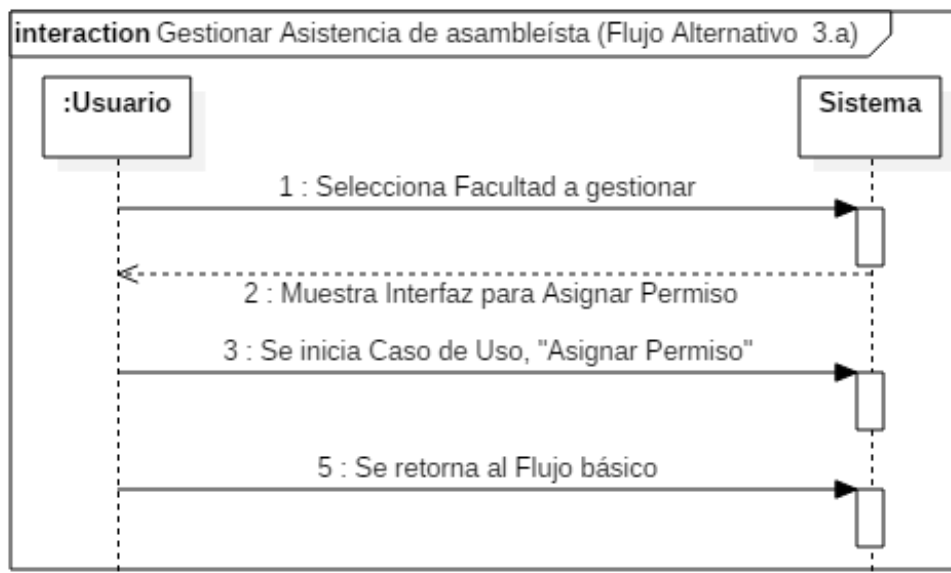


Figura 34. DSS - Gestionar Asistencia de asambleísta (Flujo Alternativo 3.a)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Asignar Permisos**

Tabla 21. Especificación CU - Asignar Permisos

Fuente: Elaboración propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Asignar Permisos
ACTOR	Directivo, Empleado (rol: Profesional Universitario)
DESCRIPCIÓN	Este caso de uso permitirá el registro de permisos temporales y/o permanentes de los asambleístas que abandonen el lugar de la sesión plenaria, estableciendo en ellos al asambleísta que le sustituirá.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El sistema muestra un formulario para gestionar generales del permiso. 2. El actor cambia parámetros de asignación de permisos²⁰ 3. El sistema muestra un mensaje de confirmación del permiso, indicando el nombre del asambleísta que queda a cargo, según sea el caso. 4. El actor confirma el permiso
FLUJO ALTERNATIVO	N/A
PRE-CONDICIÓN	<ol style="list-style-type: none"> 1. Existencia de un período 2. Haber iniciado sesión 3. Existir asistencia asociada del asambleísta saliente.
POST-CONDICIÓN	<ol style="list-style-type: none"> 1. Mensaje de éxito del permiso registrado.

i. **DSS: Flujo normal de Éxito**

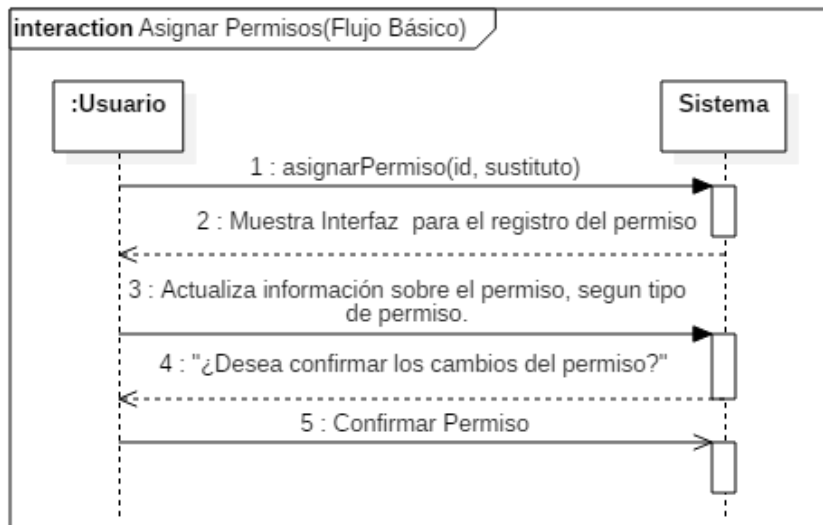


Figura 35. DSS - Asignar Permisos (Flujo Básico)

Fuente: Elaboración Propia, StarUML

²⁰ Reconsiderará que establecido el permiso Permanente, el Asambleísta no podrá registrar su ingreso posterior, además, el permiso temporal, almacenará la hora de retorno, según sea el reingreso.

- **Caso de Uso: Administrar Sesiones Plenarias**

Tabla 22. Especificación CU -Administrar Sesiones Plenarias
Fuente: Elaboración Propia.

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Administrar Sesión plenaria
ACTOR	Directivo, Empleado (rol: Profesional Universitario)
DESCRIPCIÓN	Este caso de uso permitirá, inicializar formalmente la sesión plenaria, previo a oficializar la agenda a tratar por los Asambleístas.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El sistema muestra un formulario, donde se muestran datos previamente establecidos, como fecha, hora de convocatoria, tipo de sesión(Ordinaria/Extraordinaria), así como su correlativo, y espacios para ingresar nombre de la sesión 2. El actor llena los espacios indicado y selecciona iniciar sesión. 3. El sistema lleva a cabo el caso de uso Reestructurar Agenda.
FLUJO ALTERNATIVO	N/A
PRE-CONDICIÓN	<ol style="list-style-type: none"> 1. Existencia de un período 2. Haber iniciado sesión 3. Existencia de Agenda propuesta. 4. Establecimiento de Quorum mínimo
POST-CONDICIÓN	<ol style="list-style-type: none"> 1. Mensaje de éxito, del inicio de sesión plenaria.

i. **DSS: Flujo normal de Éxito**

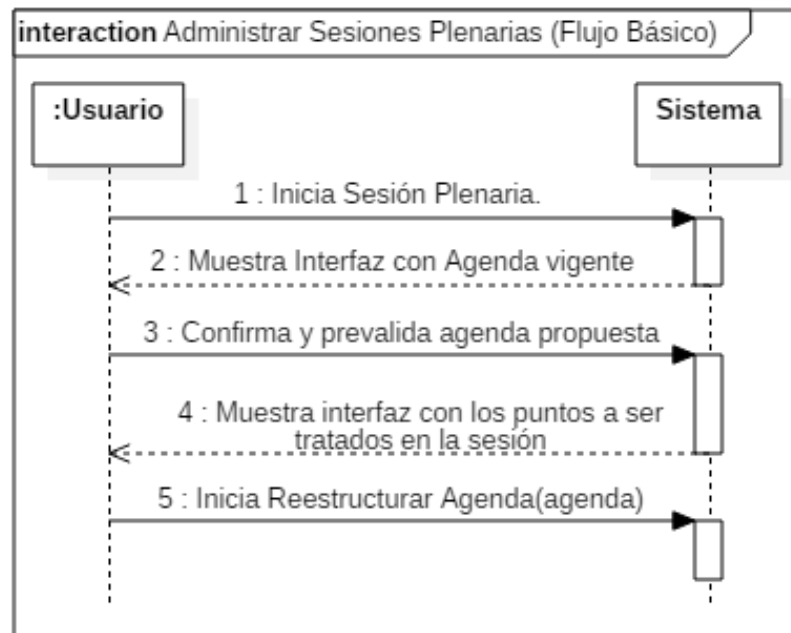


Figura 36. DSS - Administrar Sesiones Plenarias (Flujo Básico)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Gestionar Punto**

Tabla 23. Especificación CU - Gestionar Punto

Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Gestionar Punto
ACTOR	Directivo, Empleado (rol: Profesional Universitario)
DESCRIPCIÓN	Este caso de Uso permitirá, el registro de votos para la toma de acuerdos tomados por puntos discutidos en la agenda ratificada por la Asamblea General Universitaria.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El actor Selecciona el punto que será editados. 2. El sistema despliega formulario para ingresar Propuesta y/o Participación de Asambleístas 3. El actor agrega una propuesta o nombre del asambleísta a participar.²¹ 4. El sistema muestra las casillas correspondientes a los tres tipos de votación existentes (Favor, contra, abstención). 5. El actor registra los resultados obtenidos, las veces que sean requeridas según el punto. 6. El sistema muestra si el punto fue aprobado o no de acuerdo al resultado ingresado. 7. El actor puede agregar asambleísta, si este razona su voto como “abstención” o “en contra”
FLUJO ALTERNATIVO	N/A
PRE-CONDICIÓN	<ol style="list-style-type: none"> 1. Existencia de un período 2. Haber iniciado sesión 3. Haberse reestructurado agenda
POST-CONDICIÓN	<ol style="list-style-type: none"> 1. Los datos ingresados por puntos, se agregan a plantilla de acuerdo y acta, según corresponda.

²¹ Indicar el asambleísta a participar, permite una edición más rápida de las participaciones a la hora de trabajar el acta de la sesión plenaria, donde esta última no se genera si no se aprueba agenda.

i. DSS: Flujo normal de Éxito

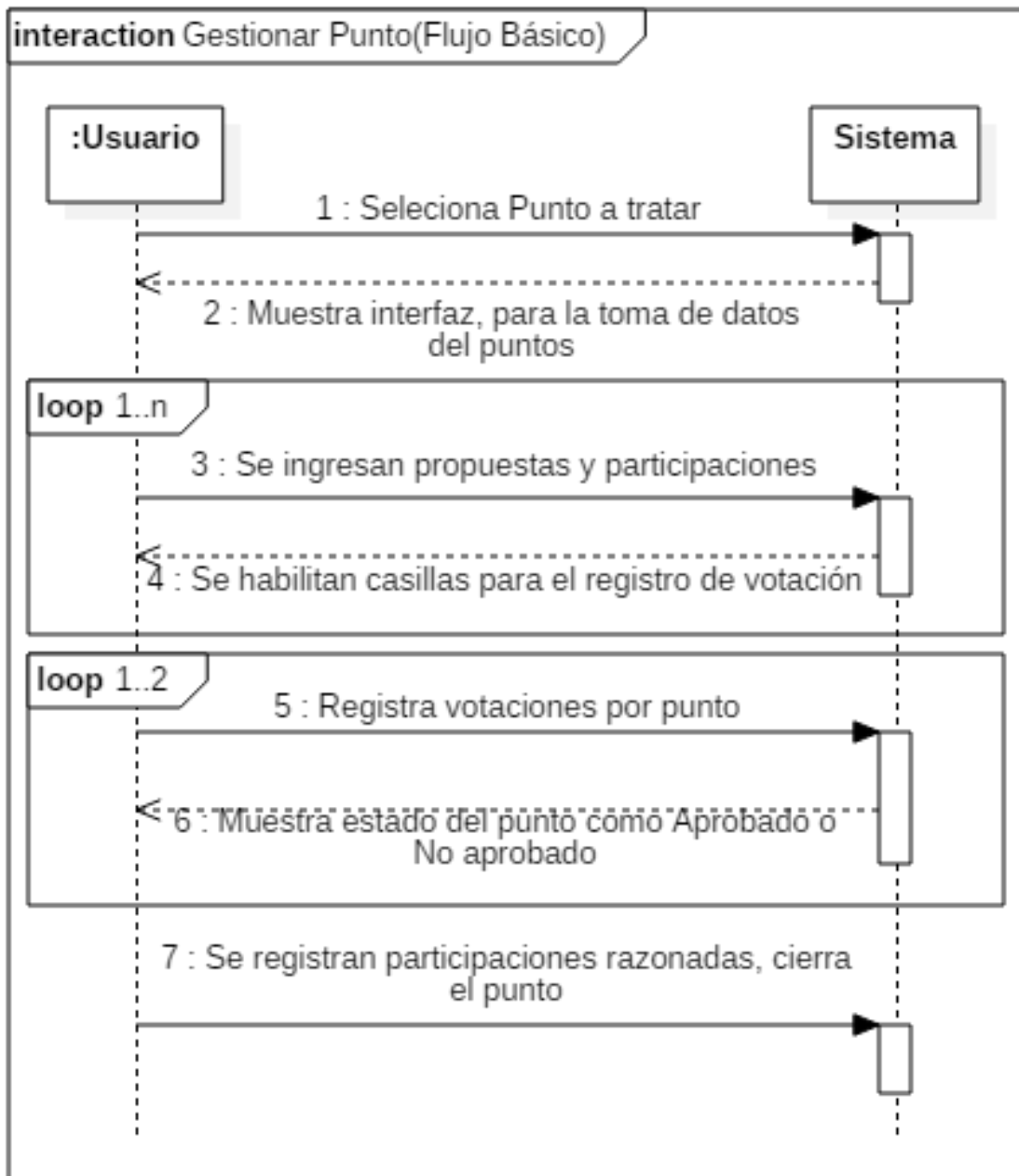


Figura 37. DSS - Gestionar Punto (Flujo Básico)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Reestructurar Agenda²²**

Tabla 24. Especificación CU - Reestructurar Agenda
Fuente: Elaboración propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Gestionar Agenda
ACTOR	Directivo, Empleado (rol: Profesional Universitario)
DESCRIPCIÓN	Este caso de uso permitirá editar la agenda sugerida por la Junta Directiva de la AGU, otorgando potestades de mover el orden de los puntos a petición de los asambleístas, así mismo, la incorporación de nuevo puntos si estos ya hubiesen sido ingresados con anterioridad al sistema
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El sistema muestra una interfaz con los ítems previamente agregados a la agenda sugerida. 2. El actor cambia el orden de los ítems existentes 3. El sistema muestra un mensaje de confirmación sobre los cambios realizados, se habilita un espacio para el registro de la votación, para aprobación de la totalidad de la agenda 4. El actor ingresa los resultados de la votación general de la agenda²³. 5. El sistema muestra un mensaje de inicio de la sesión.
FLUJO ALTERNATIVO	N/A
PRE-CONDICIÓN	<ol style="list-style-type: none"> 1. Existencia de un período 2. Haber iniciado sesión 3. Haberse iniciado Sesión 4. Poseer una agenda sugerida 5. Existencia de peticiones en el sistema
POST-CONDICIÓN	<ol style="list-style-type: none"> 1. Iniciar caso de Uso, gestionar punto

²² Análogo al Caso de Uso Gestionar Agenda, no pueden Agregarse más puntos, según Reglamento Interno de la AGU.

²³ Se da inicio de la sesión y con ella la discusión de puntos, como se muestra en el Caso de uso Gestionar Punto

- **Caso de Uso: Generar Reporte**

Tabla 25. Especificación CU - Generar Reporte
Fuente: Elaboración Propia.

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Generar Reporte
ACTOR	Directivo, Coordinador, Empleado (roles: Profesional Universitario, Técnico Financiero)
DESCRIPCIÓN	El sistema permitirá generar reportes para las diferentes funcionalidades que posee, ya sea después de una sesión plenaria, después de generar una nueva agenda, después de la asignación de dietas a los asambleístas, después de la generación de asistencia a las sesiones plenarias.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. Usuario entra al módulo de generación de reportes. 2. Escoge el reporte que desea generar. 3. Aplica los parámetros necesarios para la generación del reporte. 4. Presiona generar. 5. Se desplegará una vista previa del reporte en el navegador web. 6. El usuario definirá si imprime directamente o guarda en medios locales.
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 5.a. El documento no tiene páginas. <ol style="list-style-type: none"> 1. Se deben de especificar los datos de generación correctamente.
PRE-CONDICIÓN	<ol style="list-style-type: none"> 1. Existencia de un período 2. Haber iniciado sesión 3. Existencia de peticiones en el sistema
POST-CONDICIÓN	<ol style="list-style-type: none"> 1. Generación de vista previa del reporte seleccionado.

i. DSS: Flujo normal de Éxito

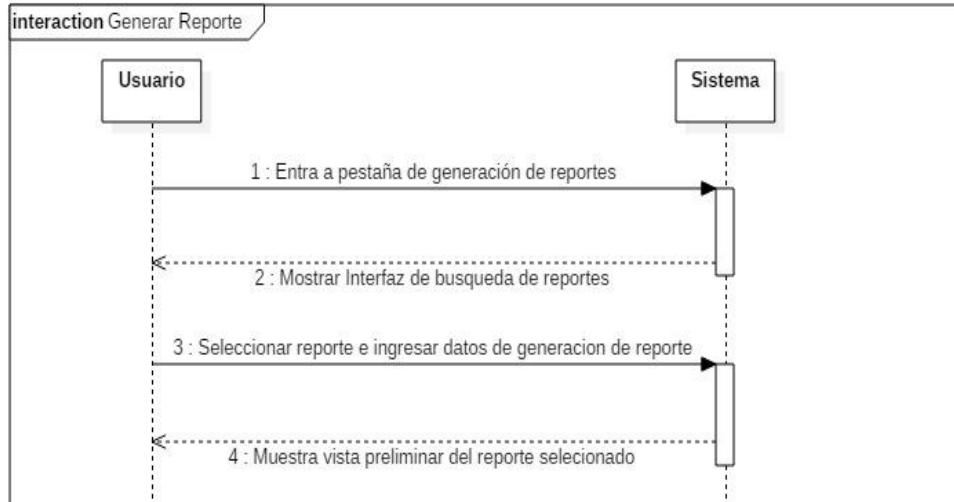


Figura 38 DSS: Generar Reporte (Flujo Básico)
Fuente: Elaboración Propia, StarUML

ii. DSS: Flujo Alternativo

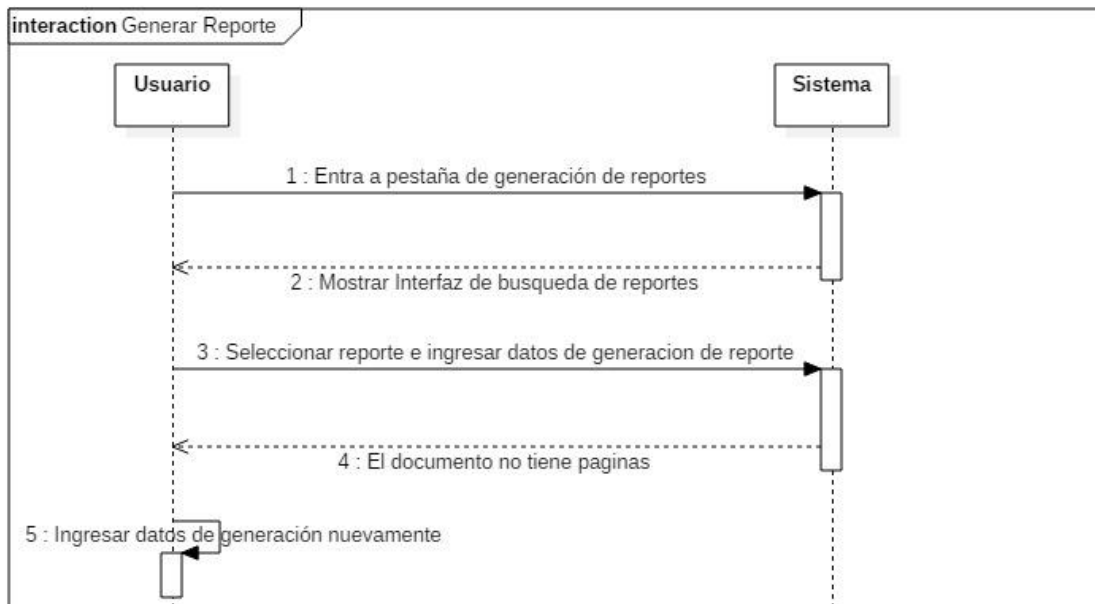


Figura 39 DSS: Generar Reporte (Flujo Alternativo 5.a)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Ingresar Documentos**

Tabla 26. Especificación CU - Ingresar Documentos
Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Ingresar documentos
ACTOR	Administrador
DESCRIPCIÓN	Este caso de uso permitirá al usuario subir al sistema la documentación oficial producida en las diferentes sesiones de la AGU, dicha documentación comprende: acuerdos, dictámenes y actas.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de Ingresar documento. 2. El sistema presenta una interfaz en la cual el usuario selecciona el archivo a subir al sistema, de igual forma el sistema presenta una caja de texto para que el usuario proporcione una breve descripción del contenido del documento a subir. 3. El usuario solicita guardar los cambios. 4. El sistema valida los datos ingresados. 5. El sistema almacena de manera exitosa el documento y presenta un mensaje de éxito al usuario.
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 4.a. El sistema determina que los datos están incompletos o erróneos. <ol style="list-style-type: none"> 1. El sistema informa error en el ingreso de la información y solicita su ingreso. 2. Se vuelve al paso 3 del flujo normal de éxito.
PRE-CONDICIÓN	El usuario debe de haber ingresado de manera correcta sus credenciales dentro del sistema.
POST-CONDICIÓN	Los documentos se almacenan de manera exitosa en el sistema

i. DSS: Flujo normal de Éxito

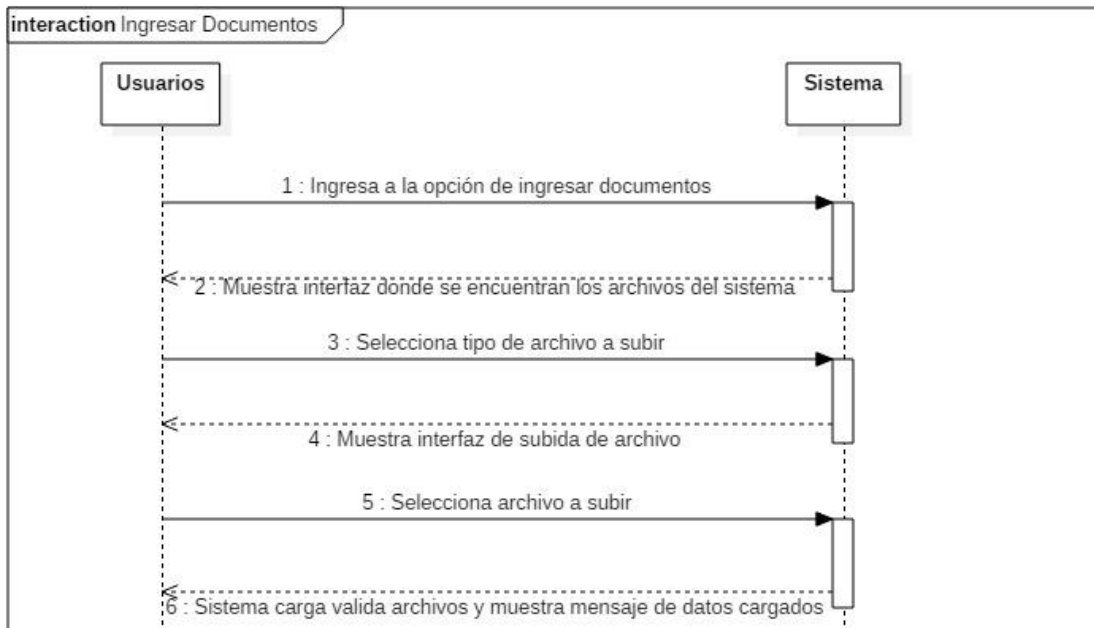


Figura 40 DSS: Ingresar Documentos (Flujo Básico)
Fuente: Elaboración Propia, StarUML

ii. DSS: Flujo Alternativo 4.a

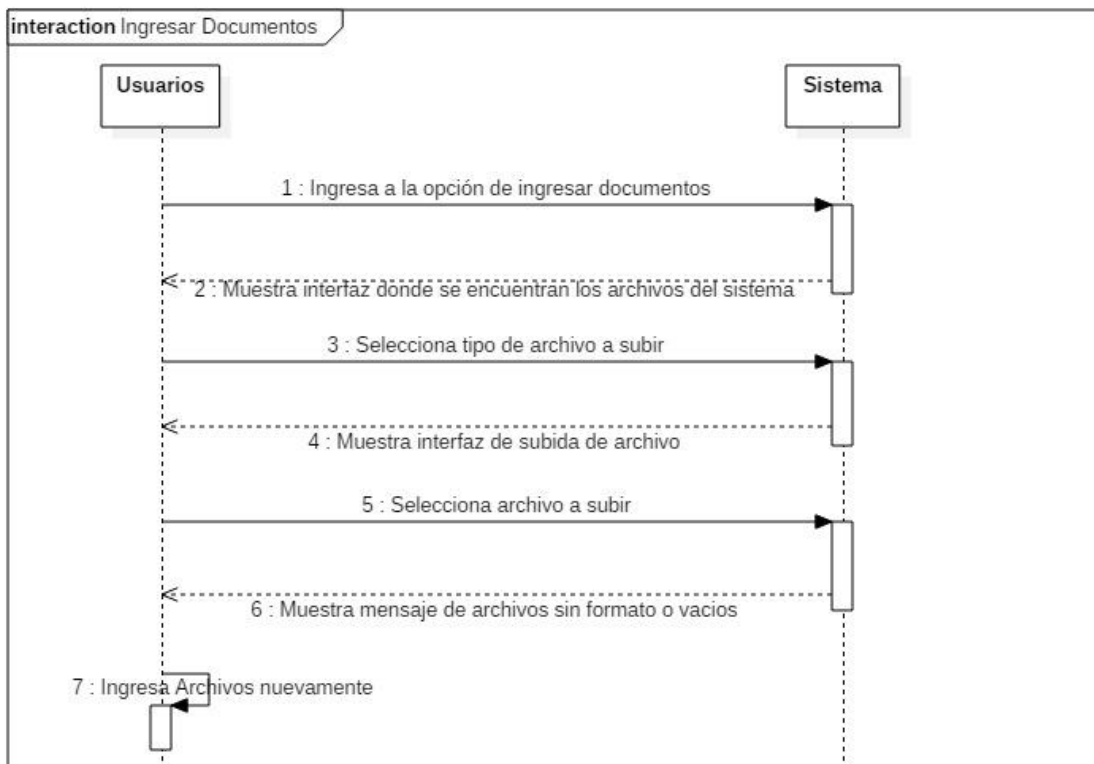


Figura 41 DSS: Ingresar Documentos (Flujo Alternativo)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Elaborar Informe Comisión**

Tabla 27. Especificación CU -Elaborar Informe Comisión
Fuente: Elaboración propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Elaborar informe comisión
ACTOR	Coordinador,
DESCRIPCIÓN	Este caso de uso permitirá al coordinador de una comisión generar la bitácora del trabajo realizado en las reuniones de trabajo, permitiendo descargar un archivo .doc, con datos importes de lo acontecido.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El Sistema muestra una interfaz con espacios para el ingreso de asuntos tratados, acuerdos tomados, actividades pendientes y observaciones. 2. El actor llena los espacios establecidos y da selecciona la opción, generar informe. 3. El Sistema descarga un documento con extensión .doc para su posterior edición si es necesario.
FLUJO ALTERNATIVO	N/A
PRE-CONDICIÓN	<ol style="list-style-type: none"> 1. Haber iniciado sesión 2. Haberse realizado convocatoria
POST-CONDICIÓN	1. El actor imprime reporte y entrega copia en físico al Asistente Administrativo.

i. **DSS: Flujo normal de Éxito**

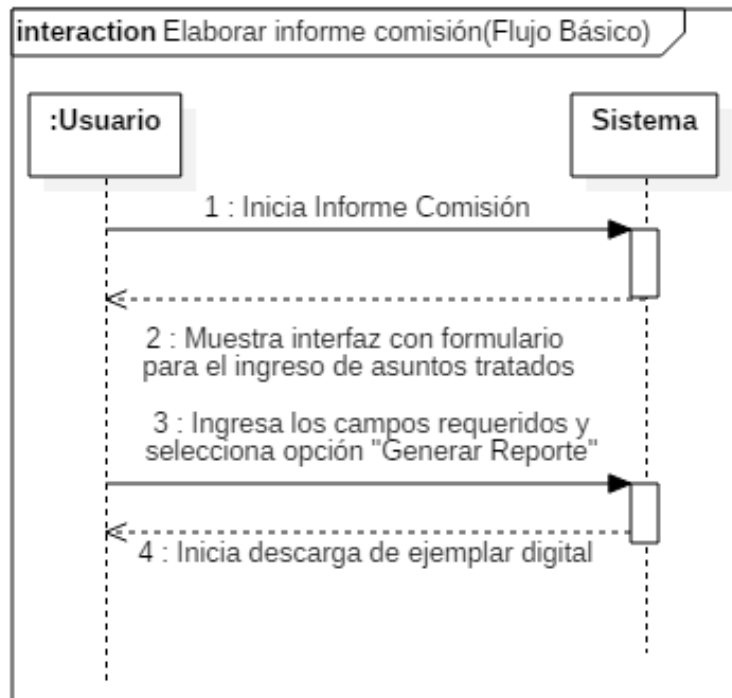


Figura 42. DSS - Elaborar informe comisión (Flujo Básico)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Consultar Documentación**

Tabla 28. Especificación CU - Consultar Documentación
Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Consultar Documentación
ACTOR	Empleado, Usuario Externo y Asambleísta
DESCRIPCIÓN	<p>El sistema permitirá al usuario consultar la documentación oficial generada por la Asamblea General Universitaria por medio de una interfaz la cual permita filtrar la documentación en base a los siguientes criterios:</p> <ol style="list-style-type: none"> 1. Rango de fechas 2. Tipo de documento (Acta, acuerdo o dictamen). 3. N° de acuerdo 4. Periodo <p>Además, permitirá descargar el documento deseado en formato PDF.</p>
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de Consultar Documentación 2. El sistema presenta una pantalla en la cual el usuario podrá filtrar los documentos presentados en base a los siguientes criterios: <ul style="list-style-type: none"> • Tipo de documento • Rango de fechas • N° de acuerdo. • Periodo de elaboración 3. El usuario selecciona una de las opciones para filtrar los documentos 4. El sistema le presenta todos los documentos que satisfacen el filtro seleccionado. 5. El usuario selecciona la opción de descargar el documento.
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 4. a. El sistema no encuentra ningún documento que satisfaga los filtros seleccionados. <ol style="list-style-type: none"> 1. El sistema informa que ningún documento se encontró en base a los criterios seleccionados por el usuario. 2. Se vuelve al paso 3 del flujo normal de éxito.
PRE-CONDICIÓN	El usuario debe de haber ingresado de manera correcta sus credenciales dentro del sistema.
POST-CONDICIÓN	El sistema descarga el documento en formato PDF

i. DSS Flujo Normal de Éxito

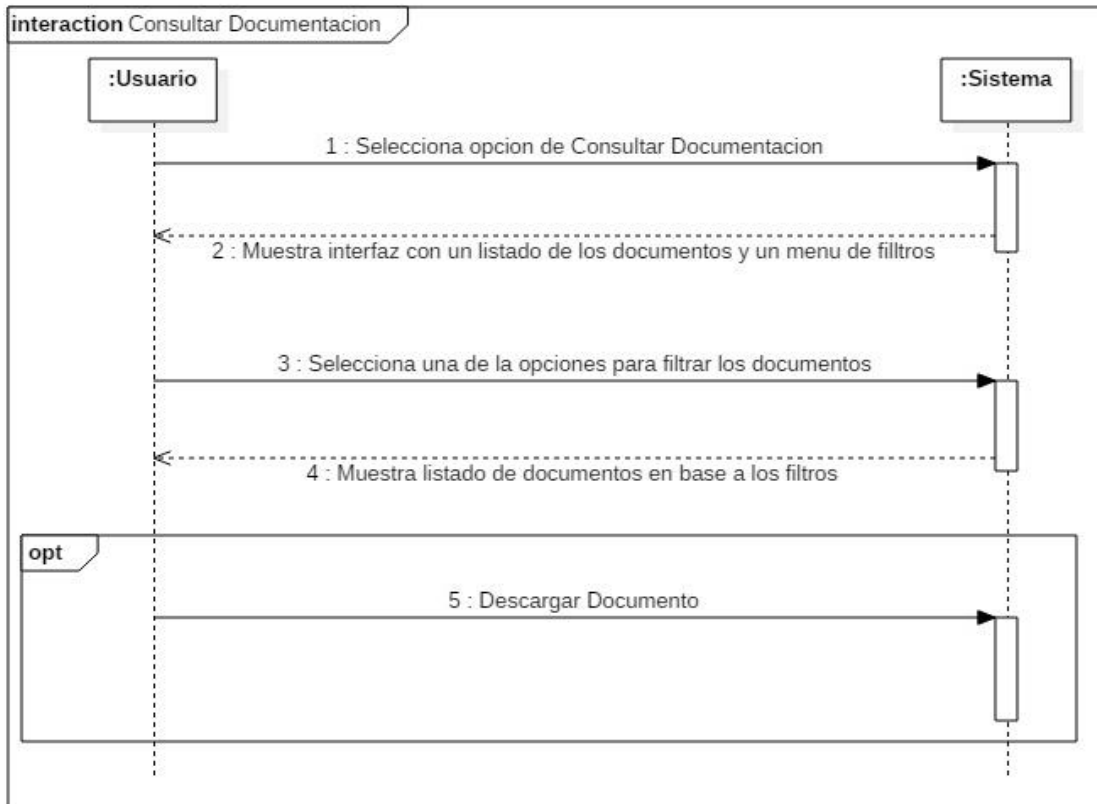


Figura 43. DSS - Consultar Documentación (Flujo Básico)
Fuente: Elaboración Propia, StarUML

ii. DSS Flujo Alternativo 4.a

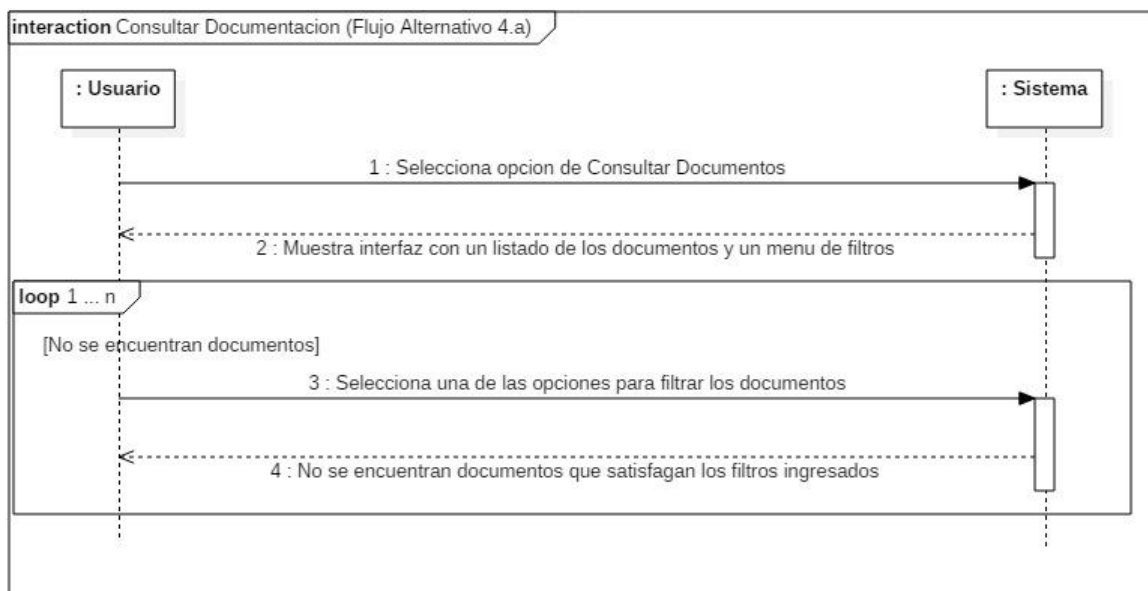


Figura 44. DSS - Consultar Documentación (Flujo Alternativo 4.a)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Modificar Plantilla**

Tabla 29. Especificación CU - Modificar Plantilla

Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Modificar Plantilla
ACTOR	Empleado, Administrador
DESCRIPCIÓN	El sistema permitirá la opción de modificar las plantillas que se muestran a los todos los que visitaran el sitio para realizar diferentes trámites.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El usuario entrara a la interfaz de modificar plantilla. 2. Se mostrarán las diferentes plantillas que están disponibles al público en general. 3. Seleccionara uno y presionara el botón de modificar. 4. Escogerá un archivo por medio de una interfaz de carga de archivos. 5. Confirmara los cambios realizados y se reemplazara la plantilla subida.
FLUJO ALTERNATIVO	<p>3.a El usuario hace clic en cancelar.</p> <ol style="list-style-type: none"> 1.1 El sistema muestra un mensaje de que la operación fue cancelada y lo redirige a la pantalla anterior. <p>5.a El actor no está de acuerdo con los cambios</p> <ol style="list-style-type: none"> 2.1 El sistema le presenta la interfaz de selección de plantillas para realizar las modificaciones que el actor considere. 2.2 Se vuelve al paso 3 del flujo básico
PRE-CONDICIÓN	<ol style="list-style-type: none"> 1. El actor debe de encontrarse autenticado en el sistema. 2. Debe de existir un documento con anterioridad si se desea modificar la plantilla.
POST-CONDICIÓN	Se modificará la plantillar con la plantilla escogida.

i. DSS: Flujo normal de Éxito

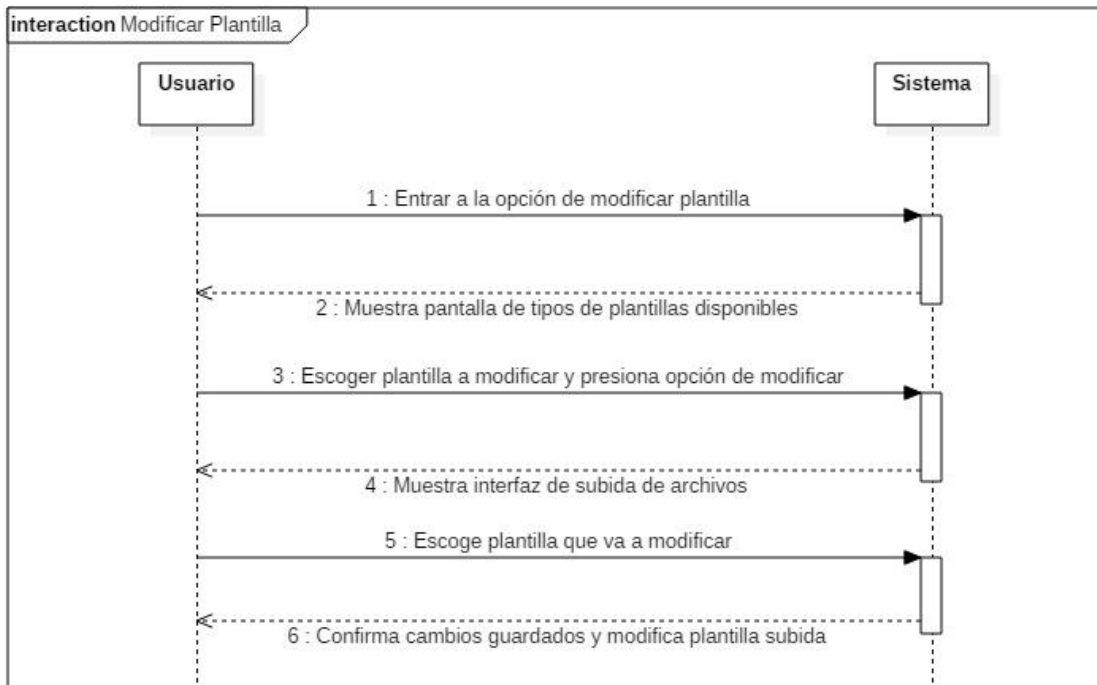


Figura 45 DSS: Modificar Plantilla (Flujo Básico)
Fuente: Elaboración Propia, StarUML

ii. DSS: Flujo Alternativo 3.a

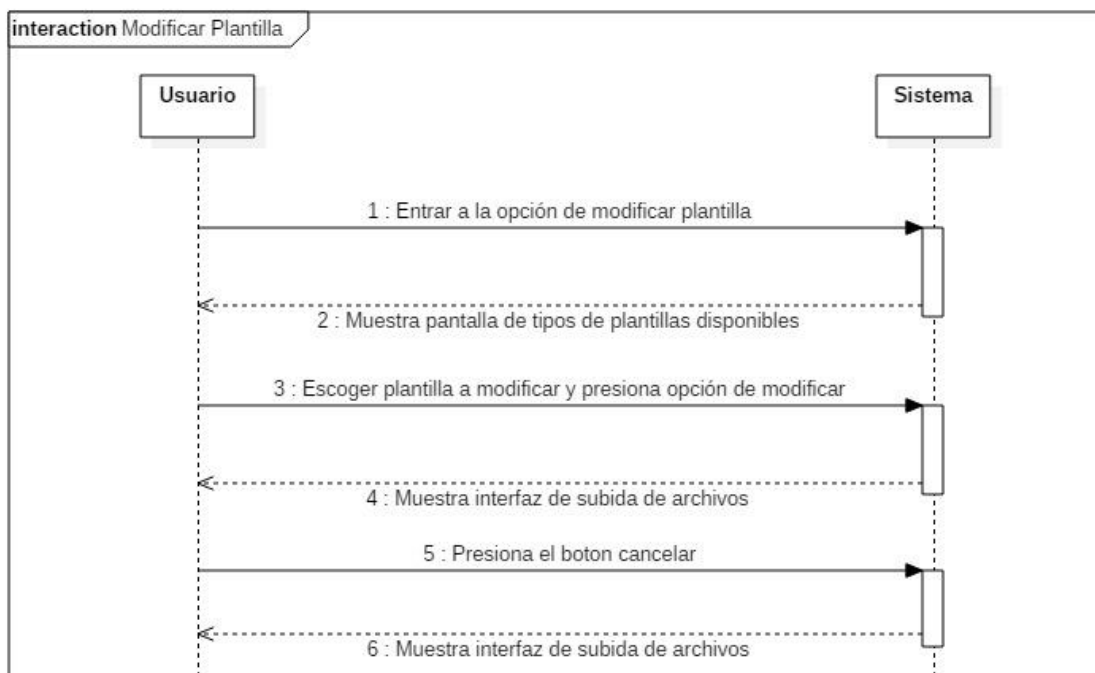


Figura 46 DSS: Modificar Plantilla (Flujo Alternativo)
Fuente: Elaboración Propia, StarUML

iii. DSS: Flujo Alternativo 5.a

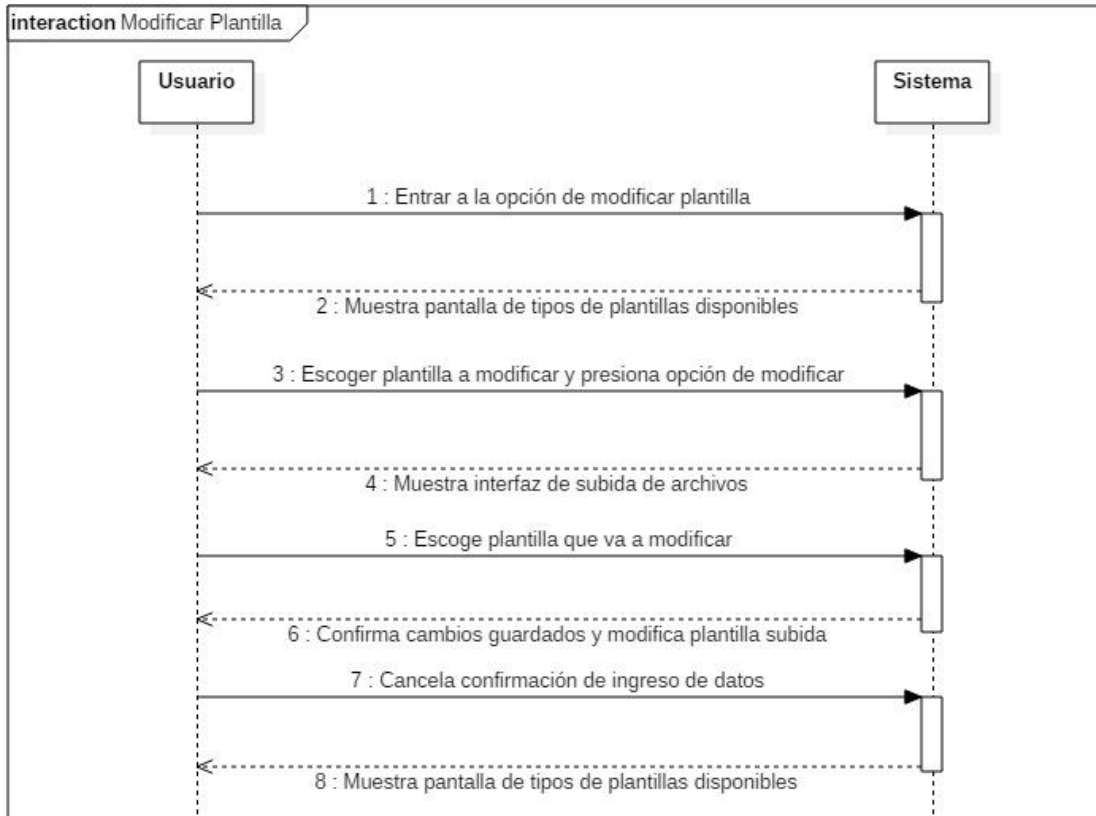


Figura 47 DSS: Modificar Plantilla (Flujo Alternativo)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Crear Período**

Tabla 30. Especificación CU - Crear Período

Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Crear periodo
ACTOR	Administrador
DESCRIPCIÓN	El sistema brindara la posibilidad de generar un nuevo periodo, tiempo en el cual la nueva Junta Directiva llevara a cabo sus funciones.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El administrador selecciona la opción de Crear un nuevo periodo. 2. El sistema presenta una interfaz en la que el usuario selecciona la fecha de inicio y fin del nuevo periodo 3. El usuario ingresa los datos y hace clic en guardar los cambios realizados. 4. El sistema consulta por medio de un mensaje si está de acuerdo con los datos ingresados. 5. El usuario acepta los cambios. 6. El sistema guarda los datos ingresados y muestra un mensaje de que la operación se llevó a cabo con éxito.
FLUJO ALTERNATIVO	<p>3.a El usuario hace clic en cancelar.</p> <ol style="list-style-type: none"> 1.1 El sistema muestra un mensaje de que la operación fue cancelada y lo redirige a la pantalla anterior. <p>5.a El actor no está de acuerdo con los cambios</p> <ol style="list-style-type: none"> 2.1 El sistema le presenta la interfaz de crear periodo con los datos anteriormente ingresados para realizar las modificaciones que el actor considere. 2.2 Se vuelve al paso 3 del flujo básico
PRE-CONDICIÓN	El actor debe de encontrarse autenticado en el sistema
POST-CONDICIÓN	Se crea el nuevo periodo.

i. DSS: Flujo normal de Éxito

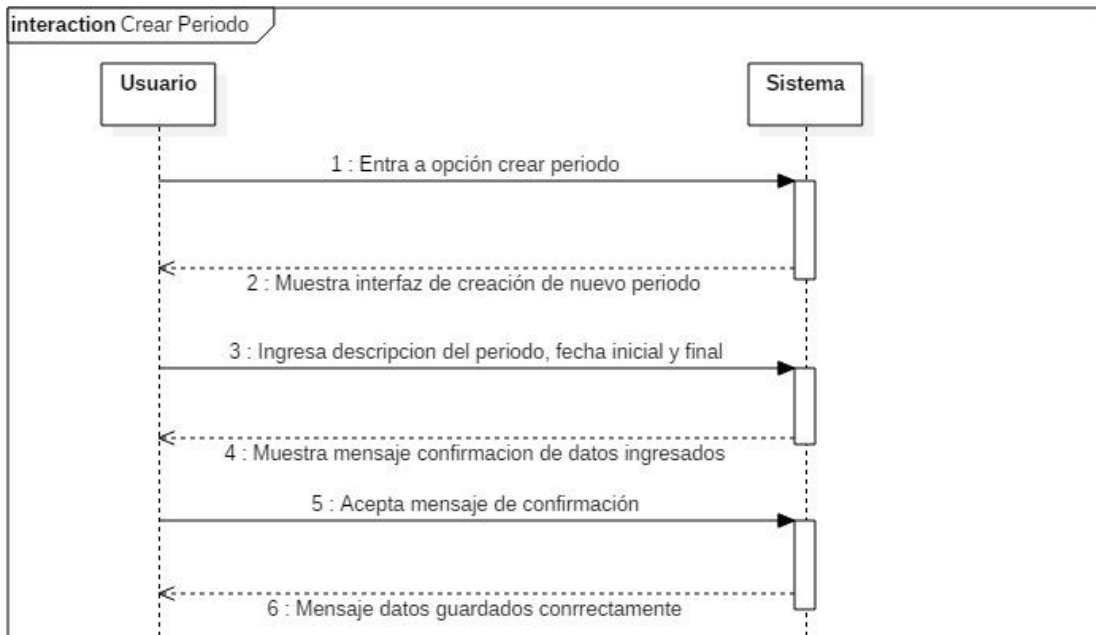


Figura 48 DSS: Crear Periodo (Flujo Básico)
Fuente: Elaboración Propia, StarUML

ii. DSS: Flujo Alternativo 3.a

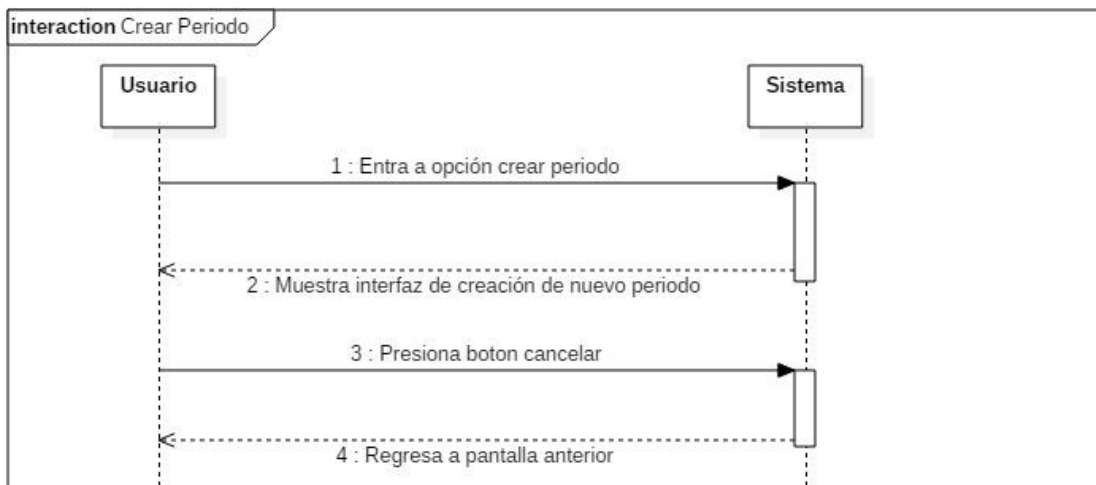


Figura 49 DSS: Crear Periodo (Flujo Alternativo)
Fuente: Elaboración Propia, StarUML

iii. DSS: Flujo Alternativo 5.a

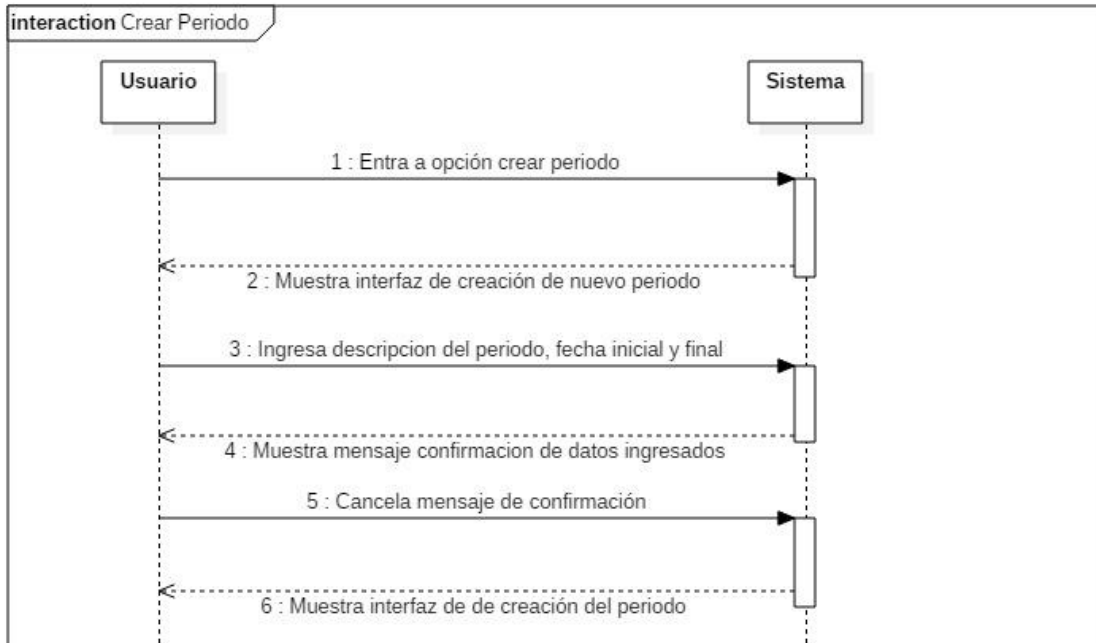


Figura 50 DSS: Crear Periodo (Flujo Alternativo)
Fuente: Elaboración Propia, StarUML

- **Caso de Uso: Establecer Parámetros**

Tabla 31. Especificación CU - Establecer Parámetros
Fuente: Elaboración Propia

ELEMENTO	DESCRIPCIÓN
CASO DE USO	Establecer Parámetros
ACTOR	Administrador
DESCRIPCIÓN	El sistema permitirá la opción de establecer los parámetros para el cálculo de las dietas de los asambleístas, apegándose a las leyes que fueron preestablecidas por la asamblea general universitaria para el cálculo de dicho bono.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. El usuario Administrador entrara a la opción de establecer parámetros. 2. Presionara la opción de modificar parámetros establecidos. 3. Elegirá cada uno de los parámetros y digitará su porcentaje o equivalencia para evaluar el bono de la dieta. 4. Presionara el botón guardar.
FLUJO ALTERNATIVO	<p>3.a El usuario hace clic en cancelar.</p> <ol style="list-style-type: none"> 1.1 El sistema muestra un mensaje de que la operación fue cancelada y lo redirige a la pantalla anterior. <p>5.a El actor no está de acuerdo con los cambios</p> <ol style="list-style-type: none"> 2.1 El sistema le presenta la interfaz de crear Establecer parámetros con los datos anteriormente ingresados para realizar las modificaciones que el actor considere. 2.2 Se vuelve al paso 3 del flujo básico
PRE-CONDICIÓN	El actor debe de encontrarse autenticado como administrador en el sistema
POST-CONDICIÓN	Se establecen los nuevos parámetros del cálculo.

i. DSS: Flujo normal de Éxito



Figura 51 DSS: Establecer Parámetros (Flujo Básico)
Fuente: Elaboración Propia, StarUML

ii. DSS: Flujo Alternativo 3.a

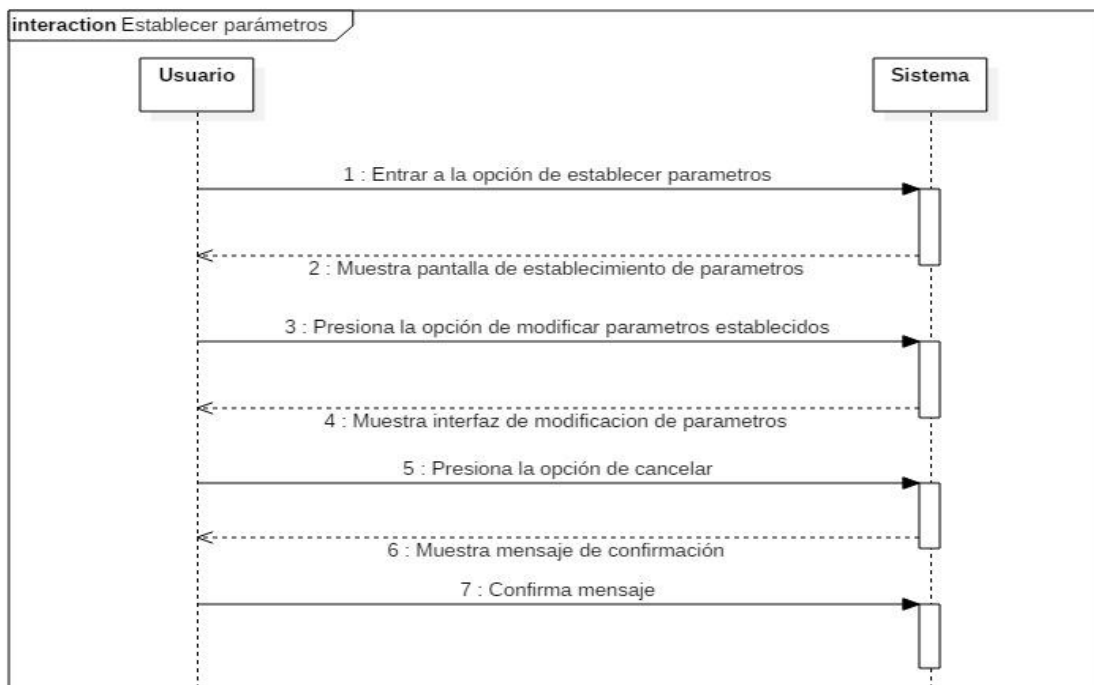


Figura 52 DSS: Establecer Parámetros (Flujo Alternativo)
Fuente: Elaboración Propia, StarUML

iii. DSS: Flujo Alternativo 5.a

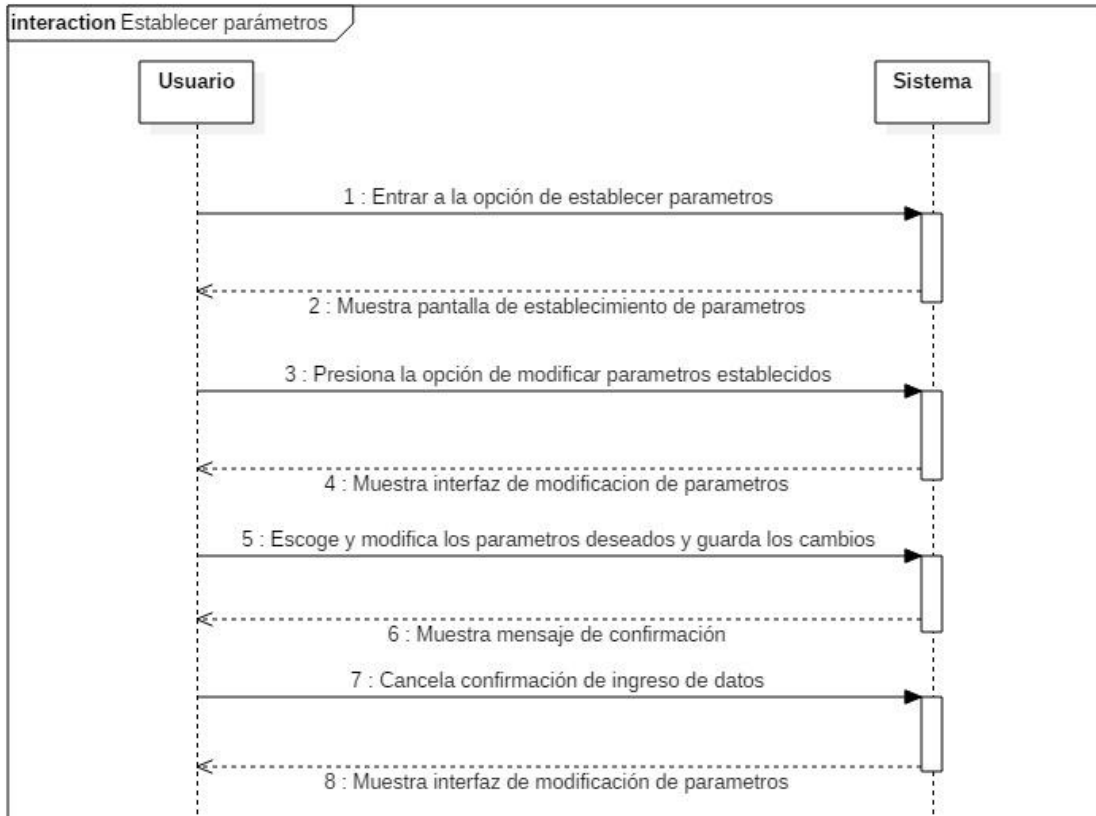


Figura 53 DSS: Establecer Parámetros (Flujo Alternativo)
Fuente: Elaboración Propia, StarUML

Capítulo II

Diseño de Sistema

2.1. Diseño de estándares

La definición de estándares en un proyecto informático es de suma importancia, debido a que son estos los que permiten obtener una mayor confiabilidad y mantenibilidad de un sistema informático, elevando la productividad y el control en la calidad del software.

Por tal razón, los estándares a definidos para el desarrollo del sistema informático de AGU son:

- ✓ Estándares de base de datos.
- ✓ Estándares de pantallas.
- ✓ Estándares de Código Fuente
- ✓ Estándares de Reportes

2.1.1. Estándares de Base de datos

Para elementos de base de datos como: nombre de las tablas, campos, llaves primarias y foráneas, triggers, funciones y procedimientos almacenados los estándares establecidos siguen los criterios a continuación descritos:

1. Tablas:

- i. Los nombres deben de ser lo más descriptivos posibles, evitando términos ambiguos o que se presten a distintas interpretaciones.
- ii. Los nombres de las tablas deben de ir escritas en minúsculas y en plural, esto es debido al ORM utilizado por Laravel, Eloquent, el cual para poder acceder a los datos de la tabla hace uso de modelos, cuyos nombres deben de corresponder a su respectiva tabla con la diferencia de que van en mayúscula y en singular. Por ejemplo:

Tabla 32. Estándar de tablas para base de datos
 Fuente: Elaboración Propia, (OTWELL, 2017)

NOMBRE EN LA BASE DE DATOS	NOMBRE DEL MODELO
USUARIOS	Usuario
PETICIONES	Peticion

- iii. Si el nombre de una tabla se compone de dos o más palabras, están irán separadas por un guion bajo “_”.
- iv. Los nombres de las tablas no deben de contener más de 64 caracteres.
- v. Los nombres no deben de iniciar con caracteres especiales, como el guion bajo “_” o el signo de dólar “\$”.
- vi. Los nombres de las tablas deben utilizar los caracteres: a-z, A-Z.

2. Campos

- i. Los campos se nombrarán usando representativo del dato y podrá estar conformado una o más palabras sin llegar a exceder a más de tres.
- ii. Usar la norma LowerCamelCase, por ejemplo: primerApellido.
- iii. Los campos clave deberán colocarse al inicio de la definición de la tabla.
- iv. La integridad referencial deberá actualizar en cascada todos los casos, y restringir el borrado salvo para las entidades débiles.

3. Llave Primaria:

- i. La llave primaria de cada tabla se identificará como id.
- ii. La llave primaria debe de ser de tipo integer.

4. Llave Foránea:

- i. Estará compuesta por el prefijo “fk”, seguido del campo clave y el nombre de la tabla a la que hace referencia, por ejemplo: fk_id_persona.

5. Triggers:

- i. Los triggers se nombrarán de manera que describan lo que realizan y se antepondrá “tr_” al nombre, podrá estar conformado de una o más palabras luego del prefijo. Por ejemplo: tr_calcular_renta.

6. Funciones:

- i. Las funciones serán nombradas de manera que describan la operación que realizan y tendrán el prefijo “fn” en su nombre, podrán estar formadas de una o más palabras luego del prefijo.

7. Procedimientos Almacenados:

- i. Para nombrar los procedimientos almacenados se utilizará el mismo estándar que para las funciones con la variación de que se utilizará el prefijo “prc” en lugar de fn.

8. Comentarios

- i. Para realizar comentarios según sea necesario en una función, trigger o procedimiento almacenado se realizarán entre una diagonal y un asterisco, siendo descriptivos y breves en lo que se quiere dar a conocer. Por ejemplo:

/* Declaración de datos */

/* Utilización de la variable X para la suma */

2.1.2. Estándares de Pantallas²⁴

El diseño²⁵ de las interfaces muestra los componentes en la pantalla del sistema, mostrados en la siguiente figura.

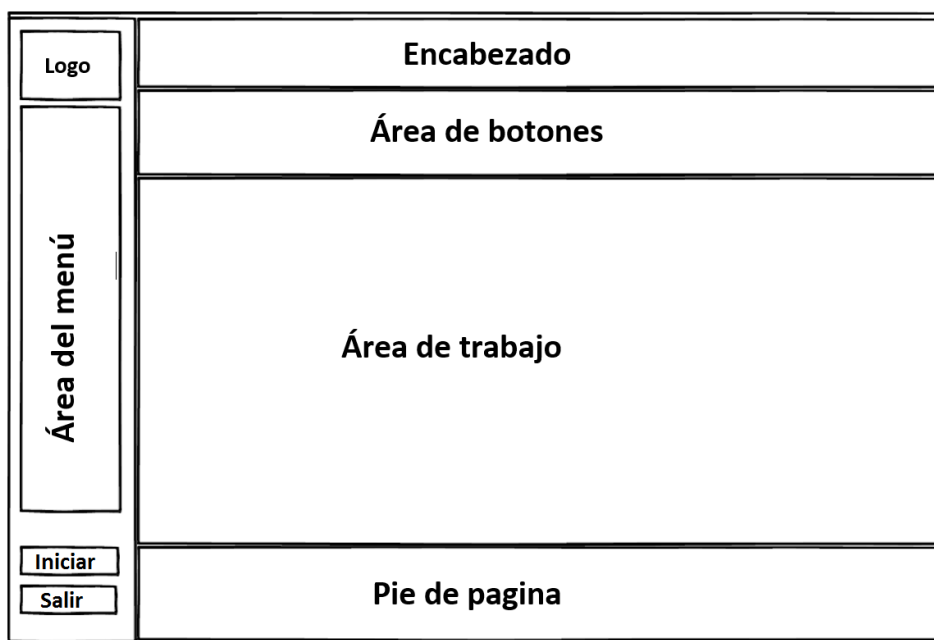


Figura 54. Diseño de estándar de pantallas
Fuente: Elaboración Propia Balsamiq Mockups 3

²⁴ Véase diseño original en anexo f) denominado: “Plantilla web SIARCA_AGU_UES”

²⁵ Diseño de interfaz general, en cuanto a secciones basado en: (SISTEMA INFORMÁTICO PARA LA ADMINISTRACIÓN DE BECAS DE POSTGRADO EN LA UNIVERSIDAD DE EL SALVADOR., 2017)

El encabezado estará formado por los siguientes elementos:

1. Título con el texto “Asamblea General Universitaria” con color rojo (código #D33F3C) y tamaño 24.
2. Subtítulo: El texto será el nombre del sistema “SISTEMA INFORMÁTICO PARA EL APOYO DE REUNIONES Y CONTROL DE ACUERDOS DE LA ASAMBLEA GENERAL UNIVERSITARIA DE LA UNIVERSIDAD DE EL SALVADOR” con color rojo (código # D33F3C) y tamaño 20.
3. Logo: Alineado a la izquierda del encabezado se encontrará el escudo de la Asamblea General Universitaria con dimensiones de 100 px de alto 100 px de ancho y abajo del mismo el texto “AGU” con color rojo (código # D33F3C) y tamaño 13.
4. Menú: El menú tendrá un fondo de color rojo (código # D33F3C) y letras blancas tamaño 13, este tendrá opciones establecidas como principales y las secundarias se mostrarán desplegándose a partir de la selección de la opción principal.
5. Rol y usuario: Se ubican en la parte derecha del encabezado con tamaño de letra 13 Arial color negro.

El cuerpo de página tendrá un fondo de color blanco, para el diseño de cada pantalla se dejarán márgenes de 20 px a los lados al igual que el superior e inferior con respecto al encabezado y pie de página.

Sobre el pie de página tenemos:

1. El pie de página tendrá dimensiones de 1075 px de ancho por 70 px de alto y un fondo de color gris (código #cccccc) y contará con los siguientes elementos:
2. Dirección: Tendrá la dirección de la universidad al lado izquierdo con letra de color negro y tamaño 13.
3. Información de contacto: Tendrá el número de teléfono de la universidad y el número de teléfono de la Asamblea General Universitaria con letra de color negro y tamaño 13

2.1.2.1. Especificaciones generales de diseño

A continuación, se describen los distintos elementos a usar en las distintas pantallas y sus especificaciones

Tabla 33. Especificaciones Generales de Diseño

Fuente: Elaboración Propia.

ELEMENTO	DESCRIPCIÓN	COLOR DE LETRA	COLOR DE FONDO
TABLAS	Deben llevar una columna para numerar las filas, una paginación con 15 filas y un control por campo para ordenar alfabéticamente.	Para las filas de color gris el texto será negro y para las filas de color azul el texto será de color blanco.	Se tendrán dos colores intercalados por cada fila y serán gris código #ddddd y azul código #597eaa.
BOTONES	Deben llevar un texto que describa con claridad la acción a realizar y un máximo de cuatro palabras.	Blanco.	Para botones primarios color azul código #597eaa, botones secundarios color rojo código #cf2a27 o color verde código #38761d.
CONTENEDORES	Deberán llevar un encabezado que describa lo que se está agrupando.	color negro para encabezados	Blanco.
TÍTULOS DE FORMULARIOS	Se utilizarán para describir el propósito de la pantalla en la que se encuentra el usuario.	color Negro código #ffffff	N/A

2.1.3. Estándares de Código Fuente

Contar con estándares a nivel de código es muy importante debido a la participación de múltiples personas en desarrollo, donde cada una podría escribir código a su manera, sin embargo, al tener un estándar a seguir, el código resulta ser claro, entendible y fácilmente modificable ya sea por cualquier miembro del equipo o por cualquier persona con los conocimientos de programación necesarios.

Entre los estándares de codificación se tomará en cuenta lo siguiente:

- 1) Se evitará el uso de la letra "ñ" para nombrar variables o funciones. En su lugar se sustituirá por la sílaba "ni". Ejemplo: año=anio.
- 2) Los nombres de variables se escribirán con inicial minúscula. Si es necesario especificar con más palabras, las demás palabras se unirán y su primera letra comienza con minúscula, utilizando el formato de escritura LowerCamelCase.
- 3) El tamaño máximo del nombre de una variable será de 25 caracteres. De ser necesario abreviar se deberá comentar el nombre completo al que representa justo a la par de su declaración.
- 4) Los nombres de las variables no podrán iniciar con caracteres especiales como guion bajo "_" o signo de dólar "\$".
- 5) El nombre de una variable sólo podrá tener de largo un carácter en los casos de que sean variables índices temporales (usadas en contadores) las cuales se ocuparán las letras "i, j, k, m, n" según sean necesarias.
- 6) Los nombres de variables constantes se escribirán en mayúscula y se deberá comentar seguido de su declaración del por qué es una constante. Por ejemplo: VALORRENTA, COLOR, etc.
- 7) Los nombres de las funciones o métodos al igual que las variables se nombrarán con letra inicial minúscula y si de ser necesario especificar con más palabras se concatenarán las subsiguientes con la primera letra mayúscula. Ejemplos: miFuncion(), sumarProduccion(anio, mes).
- 8) Los métodos de las clases deberán ser comentados indicando su función, parámetros de entrada y valor de retorno si tuviese. Los modelos, es decir, las clases que hacen alusión a las tablas de la base de datos, tendrán los mismos

nombres que de sus respectivas escritos en singular y su primera letra en mayúscula.

9) El nombre de los controladores contendrá el nombre de la entidad seguido de la palabra Controller.

10) Los comentarios deberán ser descriptivos y estar antes de que se está comentando, no usar abreviaturas.

Para comentarios cortos se usará doble diagonal y para comentarios largos se usará asterisco y diagonal. Por ejemplo:

// Comentario Corto

/* Comentario largo que se usara para describir algo que requiera más de una línea */

2.1.4. Estándares de Reportes

Como salidas del sistema se tienen los distintos reportes que serán generados. Los reportes estarán conformados del encabezado, el cuerpo del reporte y el pie de página.

Los reportes podrán tener orientación vertical u horizontal dependiendo del número de columnas que tenga.

- 1) Todos los reportes en el caso de ser requeridos su impresión serán presentados previamente en páginas de papel Bond con un tamaño carta de dimensiones 21.59 cm x 27.94 cm teniendo margen superior: 2.0 cm, margen inferior: 2.0 cm, margen izquierdo: 2.0 cm, margen derecho: 2.0 cm.
- 2) El tipo de letra será Arial con un tamaño de 12.
- 3) El color de la letra será negro y se ocupará el estilo negrito para resaltar aspectos como el título del reporte y sumalizaciones.
- 4) Incluirá el logo de la institución en la esquina superior izquierda y el nombre de la institución en la parte superior central.
- 5) Fecha en que se realizó con el formato día/mes/año (dd/MM/yyyy).
- 6) Respetar normas ortográficas.
- 7) La numeración de cada página la cual se mostrará en la esquina superior derecha con formato número de página actual/cantidad de páginas del reporte.

- 8) Indicador que permita conocer dónde es el fin del reporte.
- 9) Se mostrará el usuario que generó el reporte en la parte superior derecha.
- 10) Deberá indicarse el nombre de identificación del reporte en la parte superior derecha.
- 11) El nombre del reporte comenzará con las letras mayúsculas RPT que representará “REPORTE del sistema informático”, seguido de un correlativo, por ejemplo, RPT001, RPT002.

Dado a conocer lo anterior, la estructura de los reportes será la siguiente:

Encabezado del reporte.

- a. El encabezado tendrá el nombre de la universidad (Universidad de El Salvador), el nombre de la unidad (Asamblea General Universitaria), el nombre del reporte y su código y a la izquierda del texto el logo de la unidad de negocio.
- b. A la derecha del texto se encontrará la numeración y el nombre del usuario que generó el reporte

Cuerpo del reporte.

- a. Los reportes contendrán información pertinente según el tipo de reporte que se esté creando, todos los campos identificadores estarán en negrita.
- b. Los márgenes serán los establecidos anteriormente.

Pie de página del reporte

- a. En el pie de página se encontrará el lema de la Universidad de El Salvador: “Hacia la libertad por la cultura”.

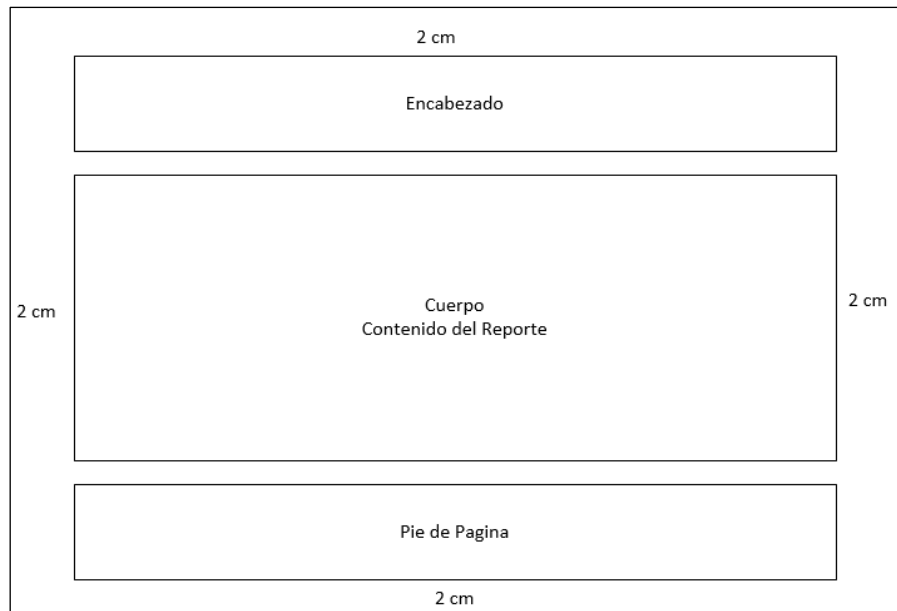


Figura 55. Diseño de pantalla de salida de datos
Fuente: Elaboración propia

2.2. Diseño de Base de Datos

El diseño de una base de datos, previo a su estructura como tal, nos ayuda a observar las diferentes entidades o tablas, así como, la forma en la que estas están relacionadas, además de ayudar en la comprensión del mundo real, para convertirlo en un modelo relacional que se pueda aplicar mediante el desarrollo e implementación de un sistema informático como se observa en la figura.

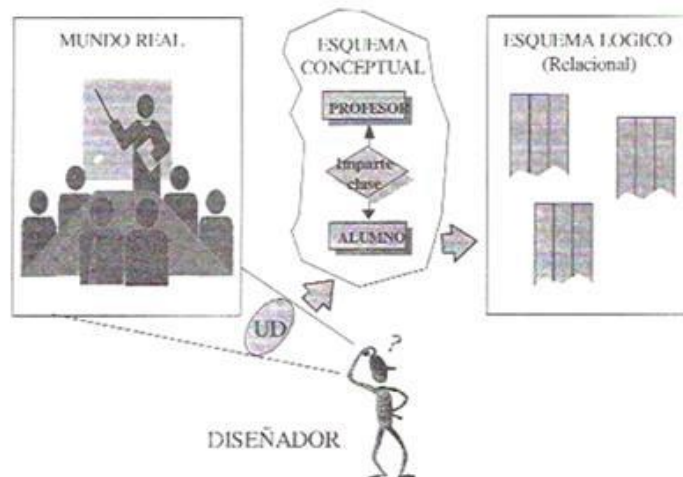


Figura 56. Diseño de base de datos
Fuente: (ALAVA, 2015)

Por lo antes expuesto, se consideró el diseño de los modelos conceptual, lógico y físico para el desarrollo del Sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de el Salvador, de los cuales encuentra referencia a continuación:

2.2.1. Modelo conceptual



*Véase en diagrama "Conceptual.svg", adjunto en CD:
/Modelos/Conceptual.svg*

2.2.2. Modelo lógico



*Véase en diagrama "Logico.svg", adjunto en CD:
/Modelos/Logico.svg*

2.2.3. Modelo Físico



*Véase en diagrama "Fisico.svg", adjunto en CD:
/Modelos/Fisico.svg*

2.3. Diseño de Pantallas²⁶

Las pantallas se refieren a la interfaz que el sistema le ofrecerá al usuario para llevar a cabo sus tareas dependiendo de su rol, la interfaz del sistema de manera general cuenta con un encabezado y un cuerpo.

A continuación, se detallan los elementos que se van a considerar para la elaboración de la interfaz gráfica de SIARCA_AGU_UES, esta será la base para elaborar las interfaces que el usuario utilizara para interactuar con él.

Tabla 34. Elementos de pantalla.

Fuente: Elaboración propia, Balsamiq Mockups

ICONO	NOMBRE	DESCRIPCIÓN
	Text box	Usado para capturar datos desde el teclado
	Button	Usado para realizar una acción dentro de la pantalla
	Calendario	Usado para delimitar rangos de fechas validos
	CheckBox	Usados para habilitar o deshabilitar elementos dentro de la pantalla, así como el cumplimiento de acciones.
	Flechas de desplazamiento	Utilizados para mover elementos dentro de una pantalla hacia arriba o hacia abajo una posición.

²⁶ Plantilla general adoptada por le SIARCA, en anexo f)

2.3.1 Búsqueda documentos

Descripción: Pantalla donde se buscará documentos almacenados dentro de la aplicación.

Encabezado

Nombre documento	fecha creacion	Opcion
Documento 1	01/01/17	Descargar
Documento 2	01/01/17	Descargar
Documento 3	01/01/17	Descargar
Documento 4	01/01/17	Descargar
Documento 5	01/01/17	Descargar
Documento 6	01/01/17	Descargar

Pie de pagina

Figura 57. Diseño de búsqueda documentos
Fuente: Elaboración Propia Balsamiq Mockups 3

2.3.2 Crear comisión

Descripción: Pantalla donde se creará nuevas comisiones y donde se podrá activar o desactivar el uso de las mismas.

Encabezado

Nombre	permanete	integrantes	estado	fecha creacion	fecha ultimo acceso
Comision de reglamentos	<input checked="" type="checkbox"/>	12		19/02/02/	01/08/17
Comision de presupuesto	<input checked="" type="checkbox"/>	29		01/02/17	20/05/17
Comision de seguimiento	<input checked="" type="checkbox"/>	15		01/02/17	20/05/17
comision temporal		15	<input checked="" type="checkbox"/>	01/02/17	20/05/17

Pie de pagina

Figura 58. Diseño de crear comisión
Fuente: Elaboración Propia Balsamiq Mockups 3

2.3.3 Administrar comisión

Descripción: Pantalla donde se podrá acceder a la pantalla para integrar usuarios a las comisiones y retirarlos. También el acceso a cada comisión para llevar el trabajo que esta realiza

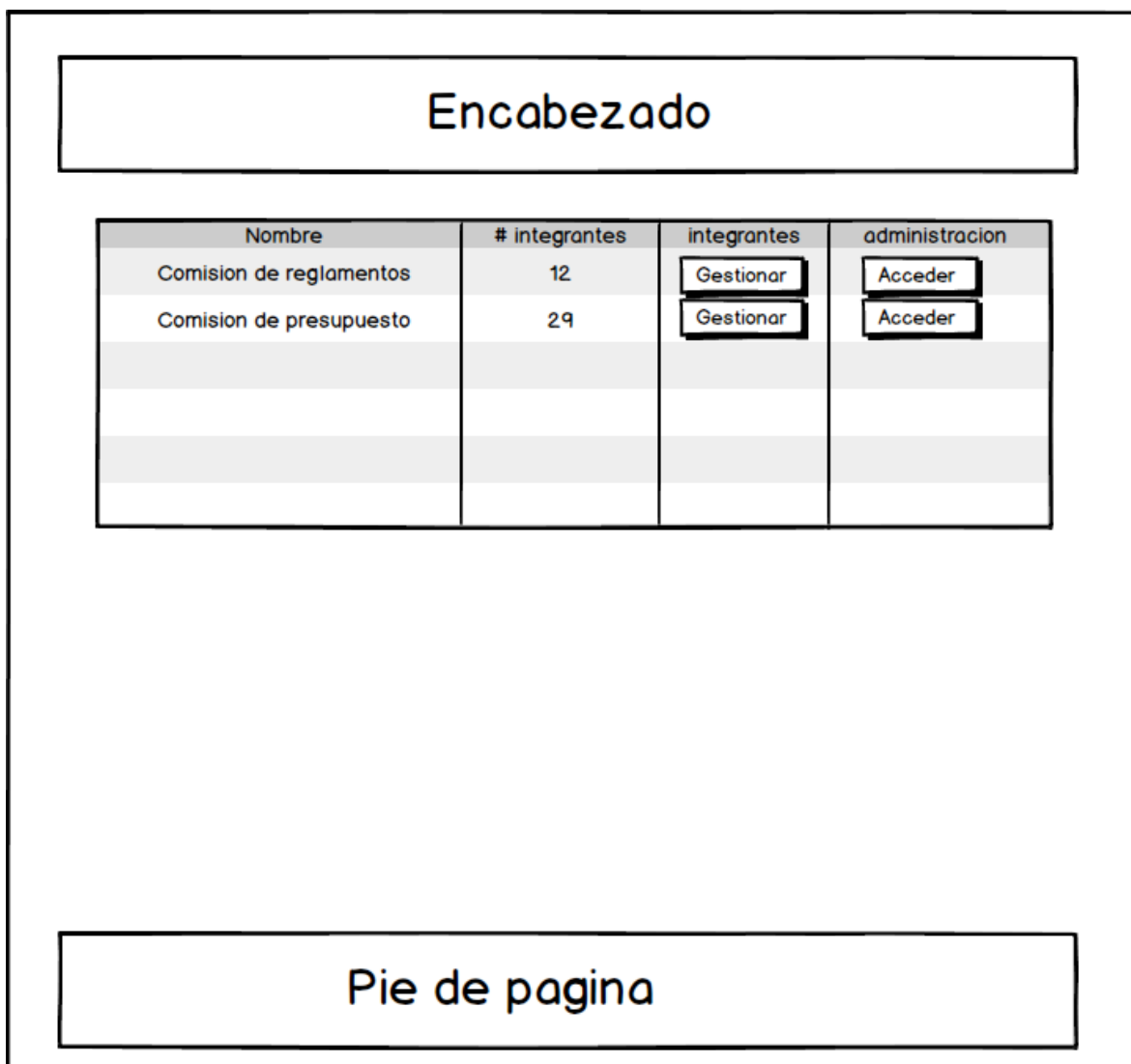


Figura 59. Diseño de administrar comisión
Fuente: Elaboración Propia Balsamiq Mockups 3

2.3.4 Administrar integrantes de comisión

Descripción: Pantalla donde se podrá integrar usuarios a las comisiones y retirarlos.
Accesible desde la pantalla Administrar Comisión.

nombre	sector	facultad	cargo	
asambleista1	docente	facultad1	propietario	retirar
asambleista2	estudiante	facultad4	suplente	retirar

Figura 60. Diseño de administrar integrantes de comisión
Fuente: Elaboración Propia Balsamiq Mockups 3

2.3.5 Trabajo de comisión

Descripción: Pantalla donde se podrá ver el trabajo por realizar en cada comisión, crear convocatorias, administrar asistencia a la comisión, así como el historial de dictámenes realizados por la misma y las bitácoras. Accesible desde la pantalla Administrar Comisión.

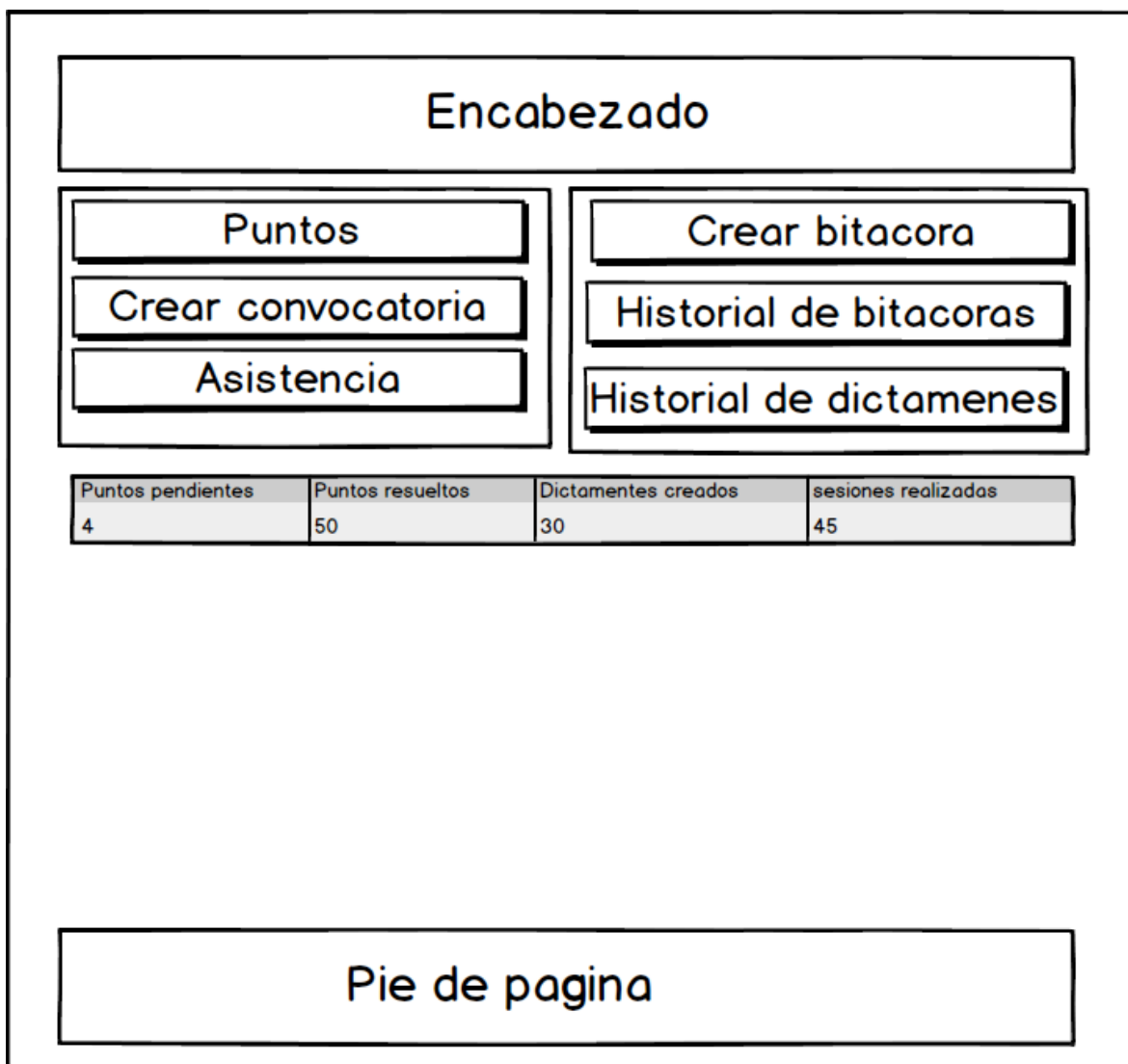


Figura 61. Diseño de trabajo de comisión
Fuente: Elaboración Propia Balsamiq Mockups 3

2.3.6 Gestionar Asistencia Asambleísta

Descripción: Pantalla donde se llevará el control de la asistencia de los asambleístas antes de iniciar ña sesión plenaria, así como, el desarrollo de la misma.

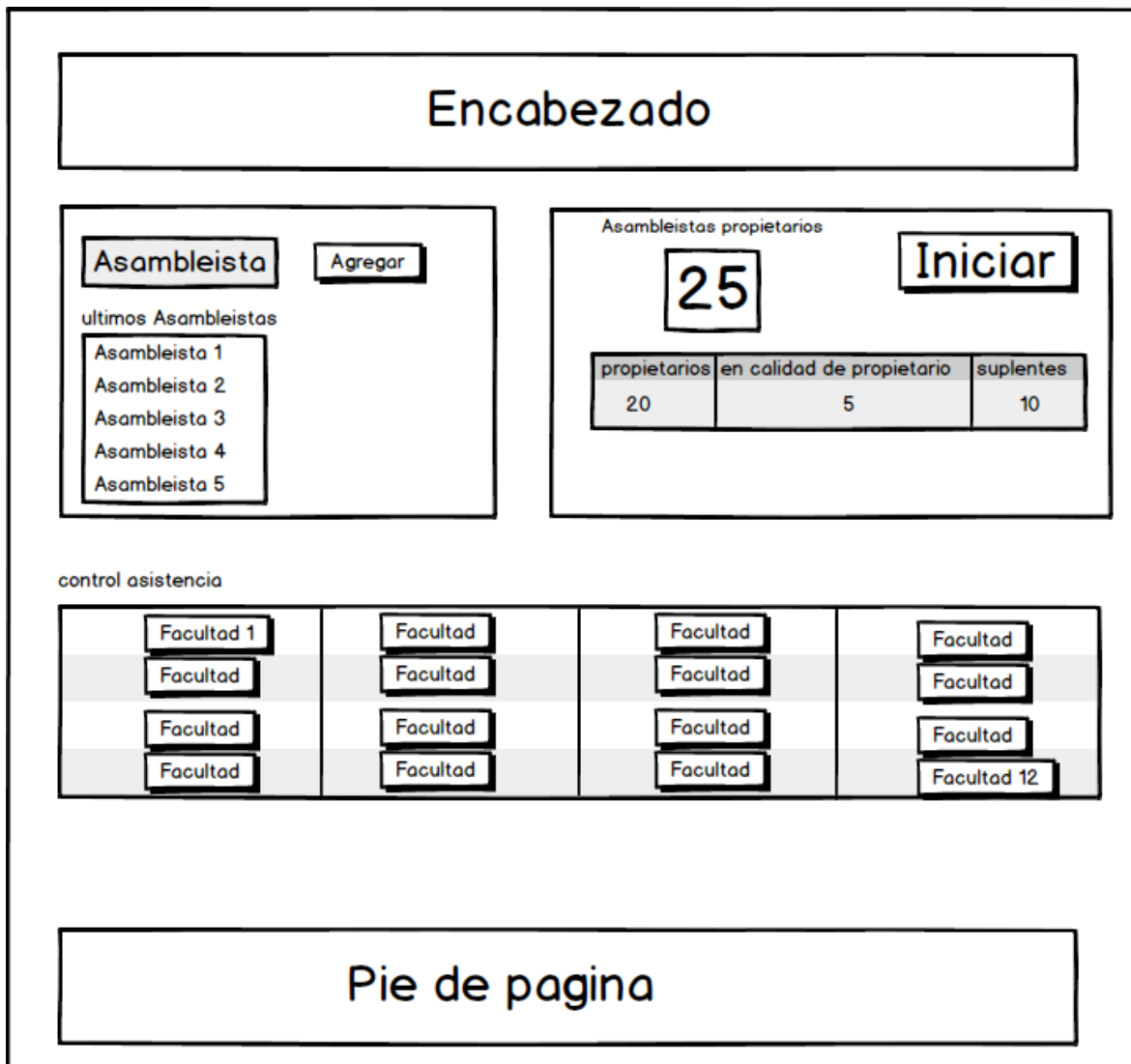


Figura 62. Diseño de gestionar asistencia de asambleísta
Fuente: Elaboración Propia Balsamiq Mockups 3

Nota: Para mayor detalle de pantallas, recomendamos consultar El manual de usuario.

2.4. Diseño de Seguridad

La información constituye un importante activo, esencial para las necesidades de una organización y puede existir de diferentes maneras, por tal razón hemos considerado de vital importancia dedicar un apartado para la seguridad de dicho elemento.

Lo que se buscó del SIARCA_AGU_UES fue asegurar la confidencialidad, accesibilidad, integridad y disponibilidad de la información y de los datos minimizando los riesgos a los que pueden estar expuestos.

Para garantizar que la información del sistema se mantenga segura, se tomó en cuenta características como.

- 1) **Confidencialidad:** La información debe estar disponible y no ser divulgada a personas, entidades o procesos no autorizados.
- 2) **Integridad y autenticidad:** Gestiona que la información del sistema permanezca inalterada a menos que sea modificada por personal autorizado, y que esta modificación sea registrada, asegurando su precisión y confiabilidad. Los datos deben ser verídicos, completos y confiables.
- 3) **Disponibilidad:** Hace referencia a que la información debe estar siempre disponible para ser procesada por las personas autorizadas.

Si alguna de estas características falla no se obtiene un sistema seguro y confiable, razón por la cual, el diseño de la seguridad permite establecer los puntos de control que deben ser clave para el manejo de la seguridad del sistema, tanto de la seguridad física, es decir de los componentes físicos que conforman el sistema; como la lógica del sistema a desarrollar, es decir todo lo que tiene que ver con los datos e información del sistema, y con ello evitar riesgos que perjudiquen el buen funcionamiento del mismo.

2.4.1. Amenazas y Vulnerabilidades

Se entiende como amenaza todos aquellos agentes capaces de explotar los fallos de seguridad, denominados puntos débiles y, como consecuencia de ellos, causar pérdidas o daños en los activos de una empresa, afectando a sus negocios.

Los activos están constantemente sometidos a amenazas que pueden colocar en riesgo la integridad, confidencialidad y disponibilidad de la información. Estas se dividen en tres grandes tipos:

- ✓ **Naturales:** Condiciones de la naturaleza y la intemperie que pueden dañar el activo, tales como el fuego, terremotos, inundaciones, etc.
- ✓ **Intencionales:** Amenazas deliberadas, fraudes, vandalismo, espionaje, robos y hurtos, etc.
- ✓ **Involuntarias:** Amenazas resultantes de acciones inconsistentes de usuarios (virus, divulgación de contraseñas).

Según la ISO 27000:2012 la vulnerabilidad es una debilidad de un activo que puede ser aprovechada por una o más amenazas.

También es definida como un fallo en el proyecto, implementación o configuración de un software o sistema operativo que, cuando es descubierta por un atacante por un atacante, resulta en la violación de la seguridad de un computador o un sistema computacional.

Entre las vulnerabilidades tenemos:

Tabla 35. Definición de vulnerabilidades
Fuente: Elaboración Propia.

VULNERABILIDADES	DESCRIPCIÓN
FÍSICAS	Ambiente en el que se almacena la información
HARDWARE	Defectos de fabricación, desactualización, mantenimiento inadecuado.
SOFTWARE	Aplicaciones que permiten acceso indebidos
HUMANAS	Daños que las personas pueden causar a la información.
NATURALES	Condiciones de la naturaleza que pueden provocar riesgo.

VULNERABILIDADES	DESCRIPCIÓN
ALMACENAMIENTO	Soporte físicos utilizados para almacenar la información.
COMUNICACIÓN	Fallos en la transmisión de la información.

2.4.2. Seguridad Física

La seguridad física hace referencia a la aplicación de barreras físicas y procedimientos de control, como medidas de prevención y contramedidas ante amenazas a los recursos e información confidencial. Se refiere a los controles y mecanismos de seguridad dentro y alrededor del Centro de Cómputo, así como los medios de acceso remoto al y desde el mismo; implementados para proteger el hardware y medios de almacenamiento de datos.

Los mecanismos de seguridad física para el sistema SIARCA_AGU_UES buscan cubrir las amenazas ocasionadas tanto por el hombre como por la naturaleza del medio físico en que se encuentra ubicado el centro.

Las principales amenazas que se prevén en la seguridad física son:

1. Desastres naturales, incendios accidentales tormentas e inundaciones.
2. Amenazas ocasionadas por el hombre.
3. Disturbios, sabotajes internos y externos deliberados.

2.4.2.1. Seguridad de Hardware

El hardware es frecuentemente el elemento más caro de todo sistema informático. Por tanto, las medidas encaminadas a asegurar su integridad son una parte importante de la seguridad de cualquier organización, ya que suelen poseer entre sus equipos maquinas caras, desde servidores con una gran potencia de cálculo hasta routers de última tecnología, pasando por modernos sistemas de transmisión de datos como la fibra óptica.

Son variadas las amenazas al hardware de una instalación informática, a continuación, se presentan las más importantes y algunas soluciones para evitar daños a este activo.

Incendios.

El fuego es una de las principales amenazas contra la seguridad. Es considerado el enemigo número uno de las computadoras ya que puede destruir fácilmente los archivos de información y programas. Para proteger los equipos ante esta amenaza se debe de tener en cuenta lo siguiente:

- El área en la que se encuentran las computadoras debe estar en un local que no sea combustible o inflamable.
- El centro de cómputo no debe situarse encima, debajo o adyacente a áreas donde se procesen, fabriquen o almacenen materiales inflamables, explosivos, gases tóxicos o sustancias radioactivas.
- Las paredes deben hacerse de materiales incombustibles y extenderse desde el suelo al techo.
- No debe de estar permitido fumar en el área de trabajo.
- La temperatura no debe sobrepasar los 18° C y el límite de humedad no debe superar el 65% para evitar el deterioro.
- Los centros de cómputo deben de estar previstos de equipo para la extinción de incendios en relación al grado de riesgo y la clase de fuego.
- Deben instalarse extintores manuales (portátiles) y/o automáticos (rociadores).

Inundaciones.

Las inundaciones son una de las causas de mayores desastres en centros de cómputo. Para prevenir este tipo de amenazas, se pueden emplear las siguientes medidas:

- Construir un techo impermeable para evitar el paso de agua desde un nivel superior y acondicionar las puertas para contener el agua que bajase por las escaleras.
- Instalación de un suelo falso por encima del suelo real
- Ubicar los equipos con una cierta elevación respecto al suelo.

Terremotos

Los sismos y los terremotos son uno de los desastres naturales que ocurren con mayor frecuencia en nuestro país, por lo tanto, ciertas medidas que se pueden emplear para minimizar sus daños o evitarlos por completo son:

- No situar equipos en ubicaciones muy elevadas.
- No colocar objetos pesados sobre los equipos para evitar que caigan sobre ellos.
- Separar los equipos de las ventanas para evitar que caigan sobre ellas o que objetos lanzados desde el exterior los dañen.
- Utilizar fijaciones para elementos críticos, tales como CPUs, monitores o routers.
- Situar los equipos sobre plataformas de goma, de forma que absorba la mayor parte de los movimientos protegiendo así los cabezales de los discos duros.

2.4.2.2. Control de Acceso

El control de acceso no solo requiere la capacidad de identificación, sino también asociar la apertura o cierre de puertas, permitir o negar acceso, basado en restricciones de tiempo, área o sector dentro de una organización. Es decir, se verificará el nivel de acceso de cada persona mediante aplicación de barreras (llave, tarjeta, Password, etc.).

Debido a que es posible que personal ajeno a la organización desee acceder a las instalaciones donde se encuentra ubicada la información es recomendable seguir lo siguiente:

- a. Todo personal externo deberá de presentar un documento con foto y llenar un formulario en donde se le solicite el nombre, firma y hora de llegada y salida
- b. El visitante debe ir siempre acompañado por uno o dos miembros de la organización y brindarle una identificación de visitante si es posible.
- c. Cuando el visitante se retire debe de registrar su hora de salida en el formulario antes mencionado.

2.4.3. Seguridad Lógica

Además de tener en cuenta la seguridad física, es de igual importancia tener en cuenta la seguridad lógica.

La seguridad lógica consiste en la "aplicación de barreras y procedimientos que resguarden el acceso a los datos y sólo se permita acceder a ellos a las personas autorizadas para hacerlo."

La seguridad Lógica emplea por el sistema SIARCA_AGU_UES es:

- Control de acceso
 - Control de acceso interno
 - Control de acceso externo
- Identificación y Autenticación
- Método de Cifrado
- Bitácoras
- Copias de seguridad

2.4.3.1. Controles de acceso

Los controles de acceso están diseñados para salvaguardar la integridad de la información que un sistema informático posee, para resguardar la información confidencial de accesos no autorizados. Existen dos tipos de controles de acceso: interno y externo al sistema.

2.4.3.1.1. Controles de acceso Interno

Dentro del sistema SIARCA_AGU_UES existirán roles de usuario que controlarán el acceso a la información según los privilegios de cada uno de los roles. A continuación, se presentan los tipos de accesos al sistema según el rol del usuario.

Tabla 36. Descripción de Roles del Sistema
Fuente: Elaboración Propia, SIARCA_AGU_UES.

ROL	TIPO DE ACCESO
DIRECTIVO	Este tipo de usuario tendrá acceso a todo lo relacionado con las sesiones plenarias, es decir, gestionar la agenda que será tratada en la sesión plenaria de la semana.

ROL	TIPO DE ACCESO
	<p>Sera capaz de gestionar la asistencia de los asambleístas que atienden a la convocatoria a la sesión plenaria, en caso de que estos se retiren por una u otra razón, este rol se encarga de asignar los permisos de retirarse del desarrollo de la sesión plenaria.</p> <p>Antes de comenzar a analizar cada punto de la agenda, este rol podrá reestructurar que puntos serán tratados primeros.</p> <p>El directivo también cuenta con la facultad de generar uno o varios de los reportes que la aplicación ofrece.</p>
COORDINADOR DE COMISIÓN	<p>El coordinador de comisión tendrá la facultad de administrar todo lo referente a la comisión que es coordinador, es decir, generar la convocatoria y enviarla a todos los miembros de esta, agregar la comisión a la que pertenece, incorporar asambleístas y por ultimo generar el informe de la comisión.</p> <p>De igual forma tendrá la capacidad de generar todos los informes que la aplicación ofrece.</p>
EMPLEADO	<p>El rol empleado podrá gestionar la agenda que se tratará en la próxima sesión plenaria, la asistencia a esta, registrar toda la correspondencia que llegue a la AGU y que será el insumo para las próximas agendas, generar reportes que la aplicación proporciona y consultar la documentación oficial de la AGU.</p> <p>También puede generar reportes que la aplicación ofrece.</p>
USUARIO EXTERNO	<p>El usuario externo es todo aquel que no cuenta con un nombre dentro de la aplicación y contraseña, este podrá monitorear el estado en el que se encuentra su solicitud presentada ante la AGU.</p>
ADMINISTRADOR	<p>El perfil de administrador es el que se encargara de subir al sistema toda la documentación realizada por la AGU a través de sus diversos procesos, esta documentación cuenta con una plantilla dentro del sistema, la cual a su vez el administrador</p>

ROL	TIPO DE ACCESO
	<p>puede modificarla y subir al sistema una nueva versión de ella, también cuenta con la facultad de crear un nuevo periodo de AGU.</p> <p>Finalmente, este tipo de rol podrá gestionar los usuarios que la aplicación tendrá.</p>

2.4.3.1.2. Controles de acceso Externo

Además de contar con un control de acceso interno, es importante contar un control por parte de agentes externos, los aspectos que se ha tenido en consideración lo siguiente:

- **Acceso de Personal Contratado o Consultores**

Debido a que este tipo de personal en general presta servicios temporarios, deberá de ponerse en consideración en las políticas y administración de sus perfiles de acceso.

2.4.3.2. Identificación y Autenticación

Representa la primera línea de defensa para la mayoría de los sistemas computarizados, permitiendo prevenir el ingreso de personas no autorizadas. Es la base para la mayor parte de los controles de accesos y para el seguimiento de las actividades de los usuarios.

Se denomina **Identificación** al momento en que el usuario se da a conocer en el sistema; y **Autenticación** a la verificación que realiza el sistema sobre esta identificación.

Los usuarios del sistema SIARCA_AGU_UES a excepción del usuario invitado deben de autenticarse para poder ingresar al sistema, para ello se deben de cumplir las siguientes condiciones:

- La contraseña debe contar con una longitud de máximo 13 y mínimo 8 caracteres.

- La contraseña debe de contar con números, letras mayúsculas y minúsculas y un carácter especial.
- El usuario contará con un número limitado de intentos para ingresar su usuario y su contraseña, una vez superado ese número su cuenta se bloqueará.

2.4.3.3. Método de Cifrado

Las contraseñas serán cifradas al momento de crear el usuario, para ello se hará uso del método de encriptación llamado bcrypt.

2.4.3.4. Bitácoras

El sistema contará con la capacidad de almacenar el historial de las operaciones realizadas por cada uno de los usuarios, junto con la fecha en que se llevó a cabo.

2.4.3.5. Copias de seguridad

Las copias de seguridad son fundamentales para el respaldo de la información de todo sistema, a continuación, se listan los aspectos de seguridad a considerar para el sistema.

- El Administrador del sistema deberá realizar las copias de respaldo de la base de datos del sistema semanalmente.
- Cada copia de seguridad llevará un distintivo que será la fecha en que se realizó dicho respaldo de información.
- Los Backups se almacenarán en CD-WR o DVD-WR, dependiendo del volumen de los datos.
- Las copias de seguridad deberán ser guardadas en un lugar de acceso restringido.

2.5. Diseño de Arquitectura

Establecer el diseño de la arquitectura de un sistema es de suma importancia, debido a que este debe cumplir con los aspectos de escalabilidad e integración de nuevas funcionalidades, existen arquitecturas que no pueden cumplir con los requisitos antes mencionados, debido a esto, muchos sistemas no pueden ser adaptados a cambios en un futuro, por lo que es necesario considerar una arquitectura que permita la

incorporación de nuevas funcionalidades, en base a esto se selecciona la arquitectura Cliente – Servidor a utilizar para el desarrollo del Sistema.

2.5.1 Arquitectura Cliente – Servidor

La arquitectura Cliente – Servidor consiste básicamente en que un cliente realiza peticiones a un programa el cual es representado por la máquina servidor, la cual cumple tales peticiones y da respuesta al cliente.

- **Ventajas**

- Centralización de los Recursos e Integridad de los datos.
- Escalabilidad al poder aumentar el número de clientes y servidores.
- Fácil mantenimiento al estar distribuidas las funciones y responsabilidades entre varios ordenadores.
- Sustituye la arquitectura monolítica en la que no hay distribución.
- Aplicaciones distribuidas y separadas de manera lógica.
- La capacidad de proceso está repartida entre los clientes y servidores.

- **Desventajas**

- Existe tráfico cuando un gran número de clientes envían peticiones simultáneas al servidor.
- Cuando el servidor no está disponible, las peticiones de los clientes no pueden ser satisfechas.
- El software y hardware en el servidor son determinantes, esto puede aumentar los costos.
- El cliente no dispone de los recursos que pueden existir en el servidor.

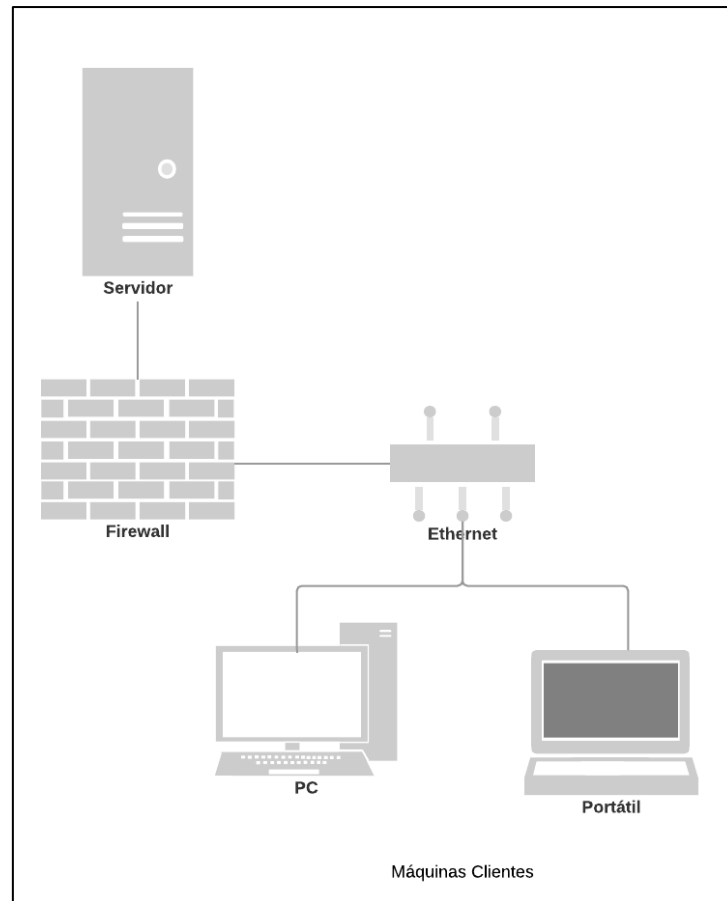


Figura 63. Arquitectura Cliente-Servidor.
Fuente: Elaboración propia / LucidChart, (ALAVA, 2015)

2.5.2 Arquitectura de la aplicación

La arquitectura a empleada en el desarrollo de la aplicación, es la arquitectura MVC (Modelo-Vista-Controlador), Laravel hace uso de ella para el desarrollo de sus aplicaciones.

El MVC o Modelo-Vista-Controlador es un patrón de arquitectura de software que, utilizando 3 componentes (Vistas, Models y Controladores) el cual separa la lógica de la aplicación de la lógica de la vista en una aplicación.

La arquitectura MVC se utiliza ampliamente en el desarrollo web, debido a que permite separar los componentes de una aplicación, logrando que un cambio en alguna parte del código, no afecte otra parte del proyecto. Por ejemplo, si se modifica la base de datos, sólo se debe modificar el modelo, mientras el resto de la aplicación permanece intacta.

Los elementos con los que cuenta la arquitectura MVC son:

- **Modelos:** Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
- **Vistas:** Es la representación visual del modelo. La vista es la responsable de generar la interfaz gráfica, basada en la información que el modelo posee.
- **Controladores:** Son los que se encargan de responder a las peticiones realizadas por el usuario y actuar sobre el modelo envía. Es el enlace entre la vista y el modelo.

Es el responsable de responder a las peticiones, actuando sobre el modelo y decidiendo que acción debe ser realizada, por ejemplo: presentar una vista o redirigir a otra página.

Adicionalmente, se encuentran los **enrutadores (routers)** los cuales mapean las rutas de una aplicación y que se encuentran designadas a una acción del controlador.

La manera en que estos elementos interactúan es el siguiente:

1. Una petición es realizada, por ejemplo, el usuario accede a una URL asociada a la aplicación.
2. Una ruta asociada con la URL mapea la URL con una acción del controlador.
3. Ese controlador invoca al modelo para recibir o actualizar la información de la base de datos y luego pasa esa información a la vista.
4. La vista es presentada al usuario por medio del navegador web.

Este proceso se puede representar con la siguiente imagen:

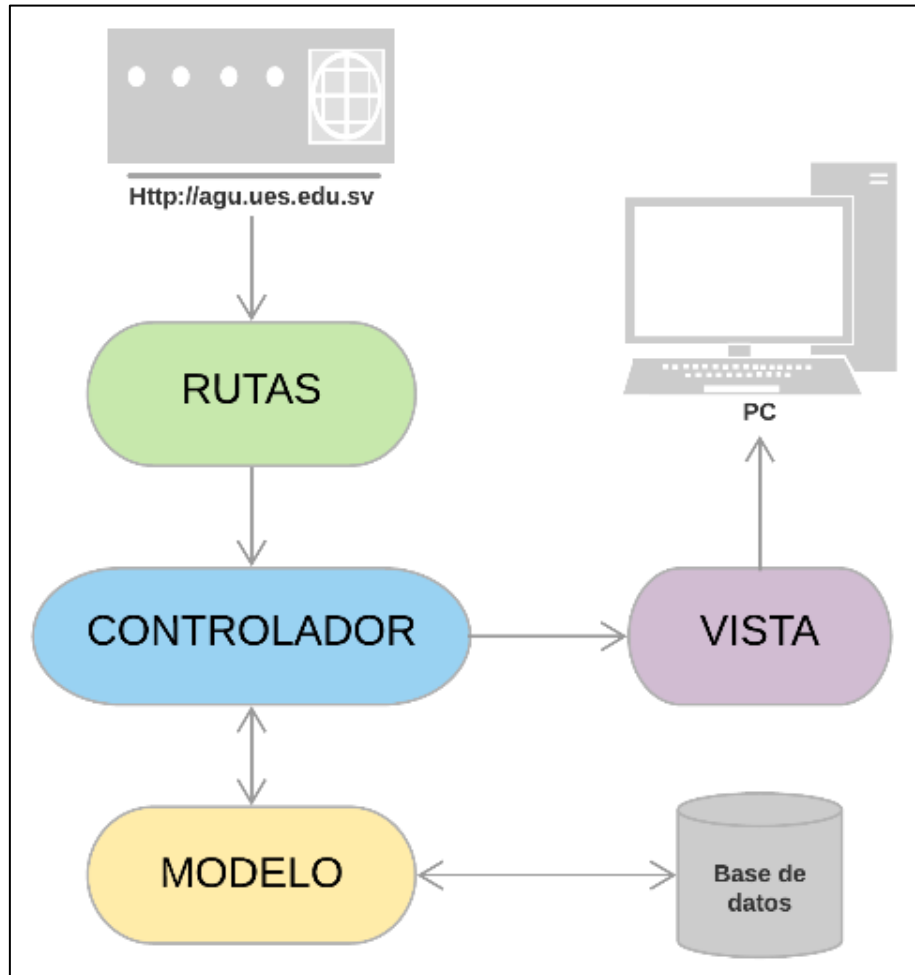


Figura 64. Modelo MVC.
Fuente: Elaboración propia / Lucid Chart, (Coleman, s.f.)

2.6. Diseño de Pruebas

2.6.1 Pruebas Unitarias

Pruebas realizadas por los programadores, cuidando que los módulos a probar no sean los desarrollados por sí mismos.

Las pruebas serán realizadas para cada uno de los módulos por separado. Evitando la interacción de los módulos entre sí. Para cada módulo el desarrollador deberá tener un banco de pruebas para el módulo en el que trabaja y deberán cumplir criterios para poder llevar una homologación entre las pruebas que se realizan a cada uno de los módulos.

Las pruebas unitarias consideradas para el SIARCA se basaron en las siguientes especificaciones:

Pruebas centradas en la verificación: Estas consistirán en determinar si estamos construyendo el sistema correctamente, a partir de los requisitos.

Pruebas centradas en la validación: Consiste en saber si estamos construyendo el sistema correcto de manera correcta, es decir si funciona de manera correcta capturando información y enviándola sin errores.

2.6.2 Pruebas integrales

A continuación, se presenta el formato sugerido para definir el modo de realización de las pruebas.

Tabla 37. Formato de pruebas integrales
Fuente: (JFernandez, 2012)

PASOS	DESCRIPCIÓN
1. IDENTIFICACIÓN	Código de prueba
2. ELEMENTOS A PROBAR	Elementos involucrados en las pruebas
3. ESTRATEGIA A SEGUIR	Estrategia a seguir para encontrar los posibles errores
4. CRITERIO DE ACEPTACIÓN	Como se sabe si la prueba ha pasado
5. CRITERIO DE SUSPENSIÓN	Cuando todas las pruebas se hayan realizado
6. TAREAS A REALIZAR PARA SATISFACER EL PROCESO	Tareas de preparación, de ejecución y de evaluación
7. NECESIDADES AMBIENTALES	Espacio de trabajo necesario para realizar pruebas
8. RESPONSABILIDADES	Miembro o miembros responsables de realizar la prueba
9. CALENDARIO	Fecha o intervalo de fecha en la cual se debe de realizar las pruebas

2.6.2.1 Pruebas de interfaces

Tabla 38. Pruebas integrales: Prueba P001
Fuente: (JFernandez, 2012)

PASOS	DESCRIPCIÓN
1. IDENTIFICACIÓN	PI001
2. ELEMENTOS A PROBAR	Módulo de administración de roles
3. ESTRATEGIA A SEGUIR	Verificar todas las vistas referente al módulo de roles
4. CRITERIO DE ACEPTACIÓN	Probar la captura de todos los datos referentes a las vistas del módulo de roles de usuario.
5. CRITERIO DE SUSPENSIÓN	Terminar las pruebas con éxito
6. TAREAS A REALIZAR PARA SATISFACER EL PROCESO	<ul style="list-style-type: none"> • Verificar todas las entradas de datos de las vistas. • Verificar que todas tienen asignadas una variable de la entidad. • Verificar que el tamaño coincida con el tamaño asignado en la base de datos. • Verificar que el tipo de dato coincida con el tipo asignado en la base datos.
7. NECESIDADES AMBIENTALES	<ul style="list-style-type: none"> • Aplicación alojada en el servidor de pruebas • Credenciales de administrador para entrar a todas las funcionalidades de las vistas
8. RESPONSABILIDADES	[Integrante responsable]
9. CALENDARIO	[Fecha]

Tabla 39. Pruebas integrales: Prueba P002
 Fuente: (JFernandez, 2012)

PASOS	DESCRIPCIÓN
1. IDENTIFICACIÓN	PI002
2. ELEMENTOS A PROBAR	Módulo de Gestión de reuniones
3. ESTRATEGIA A SEGUIR	Verificar todas las vistas referente al módulo de reuniones
4. CRITERIO DE ACEPTACIÓN	Probar la captura de todos los datos referentes a las vistas del módulo de reuniones.
5. CRITERIO DE SUSPENSIÓN	Terminar las pruebas con éxito
6. TAREAS A REALIZAR PARA SATISFACER EL PROCESO	<ul style="list-style-type: none"> • Verificar todas las entradas de datos de las vistas. • Verificar que todas tienen asignadas una variable de la entidad. • Verificar que el tamaño coincida con el tamaño asignado en la base de datos. • Verificar que el tipo de dato coincida con el tipo asignado en la base datos.
7. NECESIDADES AMBIENTALES	<ul style="list-style-type: none"> • Aplicación alojada en el servidor de pruebas • Credenciales de administrador para entrar a todas las funcionalidades de las vistas
8. RESPONSABILIDADES	[Integrante responsable]
9. CALENDARIO	[Fecha]

Tabla 40. Pruebas integrales: Prueba P003
 Fuente: (JFernandez, 2012)

PASOS	DESCRIPCIÓN
1. IDENTIFICACIÓN	PI003
2. ELEMENTOS A PROBAR	Módulo de Generación de acuerdos y dictámenes
3. ESTRATEGIA A SEGUIR	Verificar todas las vistas referente al módulo de acuerdos y dictámenes
4. CRITERIO DE ACEPTACIÓN	Probar la captura de todos los datos referentes a las vistas del módulo de acuerdos y dictámenes.
5. CRITERIO DE SUSPENSIÓN	Terminar las pruebas con éxito
6. TAREAS A REALIZAR PARA SATISFACER EL PROCESO	<ul style="list-style-type: none"> • Verificar todas las entradas de datos de las vistas. • Verificar que todas tienen asignadas una variable de la entidad. • Verificar que el tamaño coincida con el tamaño asignado en la base de datos. • Verificar que el tipo de dato coincida con el tipo asignado en la base datos.
7. NECESIDADES AMBIENTALES	<ul style="list-style-type: none"> • Aplicación alojada en el servidor de pruebas • Credenciales de administrador para entrar a todas las funcionalidades de las vistas
8. RESPONSABILIDADES	[Integrante responsable]
9. CALENDARIO	[Fecha]

Tabla 41. Pruebas integrales: Prueba P004
 Fuente: (JFernandez, 2012)

PASOS	DESCRIPCIÓN
1. IDENTIFICACIÓN	PI004
2. ELEMENTOS A PROBAR	Módulo de Generación de Reportes
3. ESTRATEGIA A SEGUIR	Verificar todas las vistas referente al módulo de Generación de Reportes
4. CRITERIO DE ACEPTACIÓN	Probar la captura de todos los datos referentes a las vistas del módulo de Generación de Reportes.
5. CRITERIO DE SUSPENSIÓN	Terminar las pruebas con éxito
6. TAREAS A REALIZAR PARA SATISFACER EL PROCESO	<ul style="list-style-type: none"> • Verificar todas las entradas de datos de las vistas. • Verificar que todas tienen asignadas una variable de la entidad. • Verificar que el tamaño coincida con el tamaño asignado en la base de datos. • Verificar que el tipo de dato coincida con el tipo asignado en la base datos. • Verificar que existen los archivos guardados en las rutas específicas.
7. NECESIDADES AMBIENTALES	<ul style="list-style-type: none"> • Aplicación alojada en el servidor de pruebas • Credenciales de administrador para entrar a todas las funcionalidades de las vistas
8. RESPONSABILIDADES	[Integrante responsable]
9. CALENDARIO	[Fecha]

Tabla 42. Pruebas integrales: Prueba P005
 Fuente: (JFernandez, 2012)

PASOS	DESCRIPCIÓN
1. IDENTIFICACIÓN	PI005
2. ELEMENTOS A PROBAR	Módulo de Control y seguimientos de solicitudes
3. ESTRATEGIA A SEGUIR	Verificar todas las vistas referente al módulo de Control y seguimientos de solicitudes
4. CRITERIO DE ACEPTACIÓN	Probar la captura de todos los datos referentes a las vistas del módulo de Control y seguimientos de solicitudes.
5. CRITERIO DE SUSPENSIÓN	Terminar las pruebas con éxito
6. TAREAS A REALIZAR PARA SATISFACER EL PROCESO	<ul style="list-style-type: none"> • Verificar todas las entradas de datos de las vistas. • Verificar que todas tienen asignadas una variable de la entidad. • Verificar que el tamaño coincida con el tamaño asignado en la base de datos. • Verificar que el tipo de dato coincida con el tipo asignado en la base datos.
7. NECESIDADES AMBIENTALES	<ul style="list-style-type: none"> • Aplicación alojada en el servidor de pruebas • Credenciales de administrador para entrar a todas las funcionalidades de las vistas
8. RESPONSABILIDADES	[Integrante responsable]
9. CALENDARIO	[Fecha]

2.6.2.2 Pruebas de configuración

Tabla 43. Pruebas integrales: Prueba P006

Fuente: (JFernandez, 2012)

PASOS	DESCRIPCIÓN
1. IDENTIFICACIÓN	PI006
2. ELEMENTOS A PROBAR	Conexión a la base de datos
3. ESTRATEGIA A SEGUIR	Verificar la configuración de conexión a la base de datos de la aplicación
4. CRITERIO DE ACEPTACIÓN	Conexión exitosa a la base de datos
5. CRITERIO DE SUSPENSIÓN	Probar que todos los módulos se puedan conectar a la base de datos
6. TAREAS A REALIZAR PARA SATISFACER EL PROCESO	<ul style="list-style-type: none">• Verificar archivo de configuración de la aplicación de conexión a la base de datos.• Verificar nombre y parámetro de conexión en la base de datos.• Verificar que todas las entidades estén creadas en la aplicación.
7. NECESIDADES AMBIENTALES	<ul style="list-style-type: none">• Aplicación alojada en el servidor de pruebas• Servicio de conexión de base de datos levantada
8. RESPONSABILIDADES	[Integrante responsable]
9. CALENDARIO	[Fecha]

Capítulo III Construcción

Tal como se describe en la sección 2.5.2 de este documento, a continuación se plasma el código relacionado a dos funcionalidades del Sistema informáticas, siendo de complejidad menor para su comprensión, por ellos, por cada funcionalidad, se presentarán, su Modelo, su vista y controlador

3.1 Inicio de sesión

El propósito de dicha funcionalidad es filtrar el acceso al Sistema informático, de acuerdo a los permisos asignados un perfil determinado

3.1.1 Modelo

El modelo que se relaciona al inicio de sesión es el mostrado en la figura siguiente.

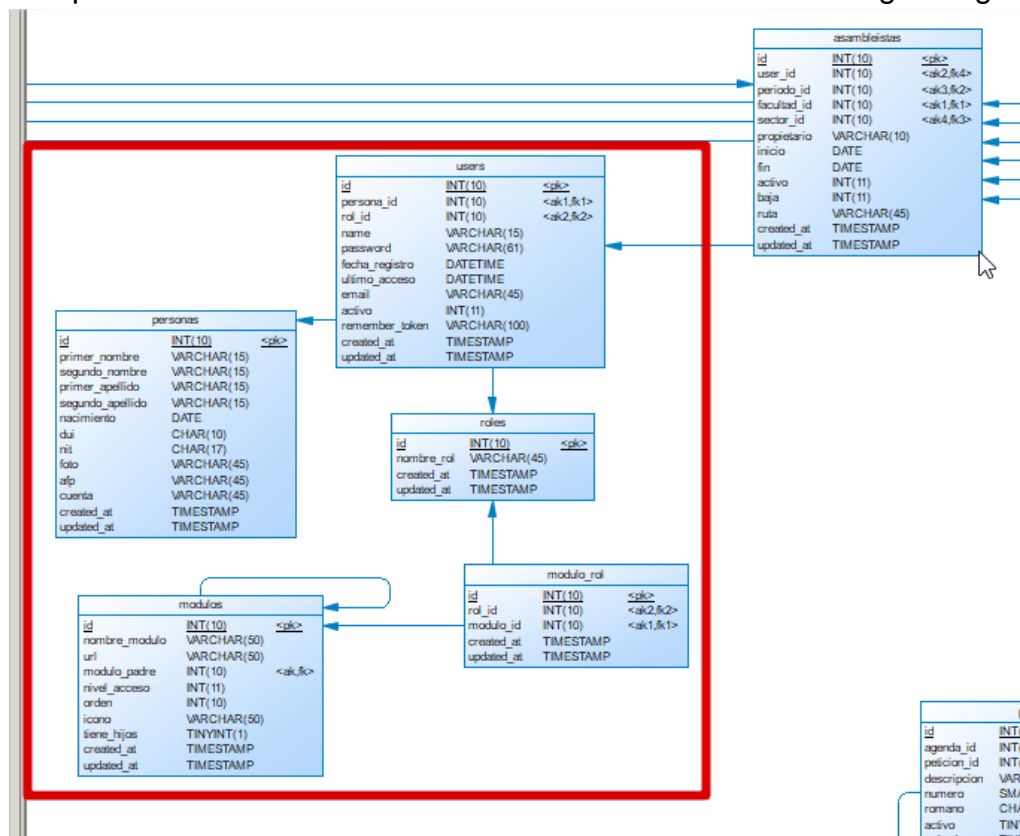


Figura 65. Modelo para el inicio de sesión
Fuente: Modelo Físico SIARCA_AGU_UES

De la imagen anterior, el código que hace posible dicha estructura por medio del framework se detalla en las imágenes a continuación indicadas.

```
1 <?php
2
3 namespace App;
4
5 use Illuminate\Foundation\Auth\User as Authenticatable;
6
7 class User extends Authenticatable
8 {
9     /**
10      * The attributes that are mass assignable.
11      *
12      * @var array
13      */
14     protected $fillable = [
15         'name', 'email', 'password',
16     ];
17
18     /**
19      * The attributes that should be hidden for arrays.
20      *
21      * @var array
22      */
23     protected $hidden = [
24         'password', 'remember_token',
25     ];
26
27     public function assembleistas()
28     {
29         return $this->hasMany('App\Asambleista');
30     }
31
32     //LLAVES FORANEAS
33     public function persona()
34     {
35         return $this->belongsTo('App\Persona');
36     }
37
38     public function rol()
39     {
40         return $this->belongsTo('App\Rol');
41     }
42
43     public function getModulosActivos(){
44         //retorna todos los modulos que coincidan con la condicion
45         return $this->getModulo()->filter(function($entry){
46             return $entry->getEstado() != 0;
47         });
48     }
49
50     /**
51      * Get permisos
52      * Este metodo retorna las rutas a las que el rol tiene permiso
53      * @return array
54      */
55     public function getPermisos(){
56         $permisos = array();
57         $modulos = $this->getModulosActivos()->toArray();
58         foreach($modulos as $modulo){
59             $permisos[] = $modulo->getUrl();
60         }
61
62         return $permisos;
63     }
64 }
65
66 }
```

Figura 66. Código para la especificación de modelo denominado "User"
Fuente: SIARCA, \App\User.php

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Persona extends Model
8  {
9      //
10     protected $table = 'personas';
11
12     public function user()
13     {
14         return $this->hasOne('App\User');
15     }
16
17
18
19 }

```

Figura 67. Código para la especificación del modelo denominado "Persona"
Fuente: SIARCA, \App\Persona.php

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Rol extends Model
8  {
9      //
10     protected $table = 'roles';
11
12     public function users()
13     {
14         return $this->hasMany('App\User');
15     }
16
17     public function modulos()
18     {
19         return $this->belongsToMany('App\Modulo', 'modulo_rol')->withTimestamps();
20     }
21
22     /*
23     public function modulo_rol()
24     {
25         return $this->hasMany('App\ModuloRol');
26     }
27     */
28
29 }

```

Figura 68. Código para la especificación del modelo denominado "Rol"
Fuente: SIARCA, \App\Rol.php

```

1 |<?php
2
3 | namespace App;
4
5 | use Illuminate\Database\Eloquent\Model;
6
7 | class ModuloRol extends Model
8 | {
9 |     //
10 |    protected $table = 'modulo_rol';
11
12 |    public function rol()
13 |    {
14 |        return $this->belongsTo('App\Rol');
15 |    }
16
17 |    public function modulo()
18 |    {
19 |        return $this->belongsTo('App\Modulo');
20 |    }
21 | }

```

Figura 69. Código para la especificación del modelo denominado "ModuloRol"
Fuente: SIARCA, \App\ModuloRol.php

```

1 |<?php
2
3 | namespace App;
4
5 | use Illuminate\Database\Eloquent\Model;
6
7 | class Modulo extends Model
8 | {
9 |     //
10 |    protected $table = 'modulos';
11
12 |    public function roles()
13 |    {
14 |        return $this->belongsToMany('App\Rol', 'modulo_rol')->withTimestamps();
15 |    }
16 |    /*
17 |    public function modulo_rol()
18 |    {
19 |        return $this->hasMany('App\ModuloRol');
20 |    }
21 |    */
22 |    public function padre()
23 |    {
24 |        return $this->belongsTo('App\Modulo', 'modulo_padre');
25 |    }
26
27 | }
28
29 | }

```

Figura 70. Código para la especificación del modelo denominado "Modulo"
Fuente: SIARCA, \App\Modulo.php

3.1.2 Vista

La vista utilizada para el ingreso al Sistema informático es la mostrada a continuación:

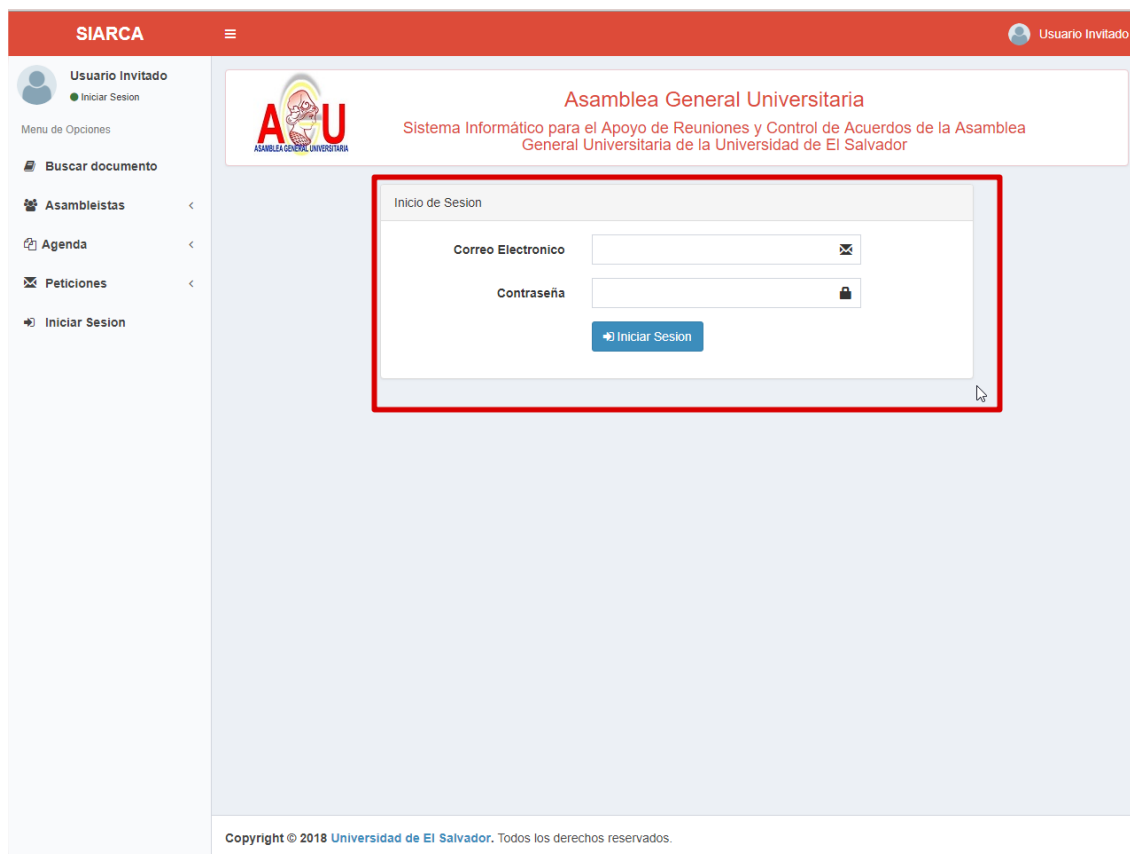


Figura 71. Vista - Iniciar Sesión
Fuente: SIARCA_AGU_UES

El código que hace posible la vista anterior se auxilia de una plantilla creada y denominada **app.blade.php**²⁷, la cual es responsable de maquetar las secciones de encabezado, pie y barra de menús, dejando a disposición, el área de trabajo para la maquetación de los formularios o elementos necesarios para determinadas vistas; por lo tanto, el código necesario para la maquetación del Login en el área de trabajo es el indicado en la figura 71.

²⁷ Puede acceder a dicho archivo en carpeta del proyecto por medio de la ruta: `resource\views\app.blade.php`

```

1  @extends('layouts.app')
2
3  @section("breadcrumb") @endsection
4  @section('content')
5  <div class="">
6  <div class="row">
7  <div class="col-md-8 col-md-offset-2">
8  <div class="panel panel-default">
9  <div class="panel-heading">Inicio de Sesión</div>
10 <div class="panel-body">
11 <form class="form-horizontal" role="form" method="POST" action="{{ url('login') }}">
12   {{ csrf_field() }}
13
14   <div class="form-group has-feedback{{ $errors->has('email') ? ' has-error' : '' }}">
15     <label for="email" class="col-md-4 control-label">Correo Electronico</label>
16     <div class="col-md-6">
17       <input id="email" type="email" class="form-control" name="email" value="{{ old('email') }}">
18       <span class="glyphicon glyphicon-envelope form-control-feedback"></span>
19       @if ($errors->has('email'))
20         <span class="help-block">
21           <strong>{{ $errors->first('email') }}</strong>
22         </span>
23       @endif
24     </div>
25   </div>
26
27   <div class="form-group has-feedback{{ $errors->has('password') ? ' has-error' : '' }}">
28     <label for="password" class="col-md-4 control-label">Contraseña</label>
29     <div class="col-md-6">
30       <input id="password" type="password" class="form-control" name="password">
31       <span class="glyphicon glyphicon-lock form-control-feedback"></span>
32       @if ($errors->has('password'))
33         <span class="help-block">
34           <strong>{{ $errors->first('password') }}</strong>
35         </span>
36       @endif
37     </div>
38   </div>
39 </div>
40
41 <!--<div class="form-group"> ...
42 </div-->
43
44 <div class="form-group">
45   <div class="col-md-6 col-md-offset-4">
46     <button type="submit" class="btn btn-primary">
47       <i class="fa fa-btn fa-sign-in"></i> Iniciar Sesión
48     </button>
49
50     <!--<a class="btn btn-link" href="{{ url('/password/reset') }}">Forgot
51     Your Password?</a-->
52   </div>
53 </div>
54 </form>
55 </div>
56 </div>
57 </div>
58 </div>
59 @endsection

```

Figura 72. Código para vista de inicio de sesión "Modulo"
Fuente: SIARCA, \resource\views\login.blade.php

3.1.3 Controlador

El controlador para el inicio de sesión, se auxilia de clases creadas por el framework para la autenticación de usuarios y cuya definición se muestra en las figuras siguientes:

```
5 use App\User;
6 use Validator;
7 use App\Http\Controllers\Controller;
8 use Illuminate\Foundation\Auth\ThrottlesLogins;
9 use Illuminate\Foundation\Auth\AuthenticatesAndRegistersUsers;
```

Figura 73. Importaciones para la autenticación de Usuarios
Fuente: Proyecto SIARCA, \ App\Http\Controllers\Auto\ AuthController.php

```
11 class AuthController extends Controller
12 {
13     /**
14      * Registration & Login Controller
15      *
16      * This controller handles the registration of new users, as well as the
17      * authentication of existing users. By default, this controller uses
18      * a simple trait to add these behaviors. Why don't you explore it?
19      */
20     use AuthenticatesAndRegistersUsers, ThrottlesLogins;
21
22     /**
23      * Where to redirect users after login / registration.
24      *
25      * @var string
26      */
27     protected $redirectTo = '/';
28
29     /**
30      * Create a new authentication controller instance.
31      *
32      * @return void
33      */
34     public function __construct()
35     {
36         $this->middleware($this->guestMiddleware(), ['except' => 'logout']);
37     }
38
39     /**
40      * Get a validator for an incoming registration request.
41      *
42      * @param array $data
43      * @return \Illuminate\Contracts\Validation\Validator
44      */
45     protected function validator(array $data)
46     {
47         return Validator::make($data, [
48             'name' => 'required|max:255',
49             'email' => 'required|email|max:255|unique:users',
50             'password' => 'required|min:6|confirmed',
51         ]);
52     }
53
54     /**
55      * Create a new user instance after a valid registration.
56      *
57      * @param array $data
58      * @return User
59      */
60     protected function create(array $data)
61     {
62         return User::create([
63             'name' => $data['name'],
64             'email' => $data['email'],
65             'password' => bcrypt($data['password']),
66         ]);
67     }
68 }
69
70
71
72 }
```

Figura 74. Clase para la autenticación de usuarios
Fuente: Proyecto SIARCA, \ App\Http\Controllers\Auth\ AuthController.php

3.2 Crear Usuario

Dicha funcionalidad permite agregar de forma individual a los usuarios del Sistema informático que tendrán acceso al mismo.

3.2.1 Modelo

El modelo que se utiliza para el registro de usuario se visualiza en la figura

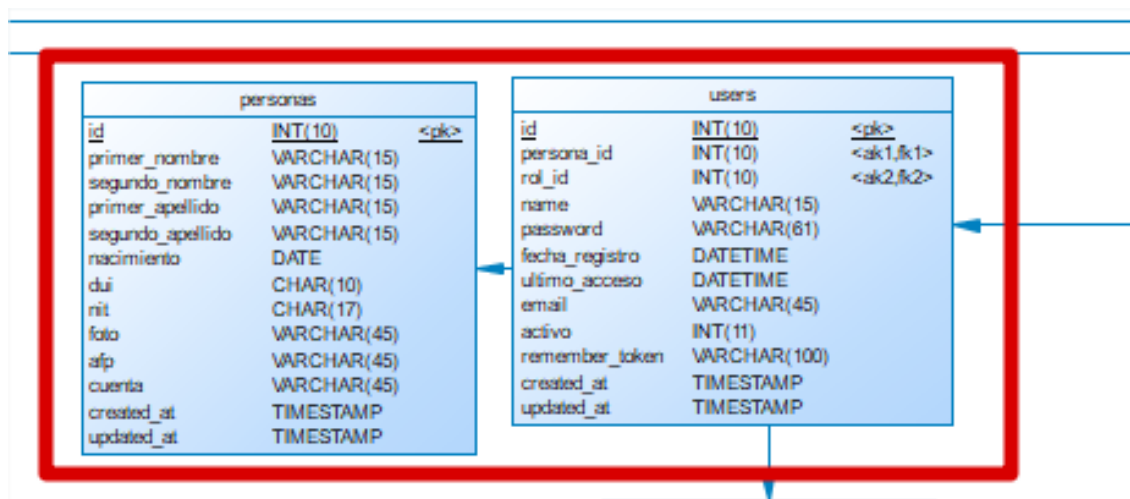


Figura 75. Modelo para el inicio de sesión
Fuente: Modelo Físico SIARCA_AGU_UES

Las especificaciones de los modelos pueden ser consultadas en las figuras 66 y 67 de estos documentos, respectivamente, debido a la coincidencia con la funcionalidad de inicio de sesión

3.2.2 Vista

La vista que hace posible el ingreso de usuarios es la indicada en la figura 76, de la cual el código html puede ser consultado desde la carpeta del proyecto en la url siguiente: *resource\views\Admisin\tracion\ RegistrarUsuarios.blade.php*, auxiliado de la pantalla indicada en la sección anterior.

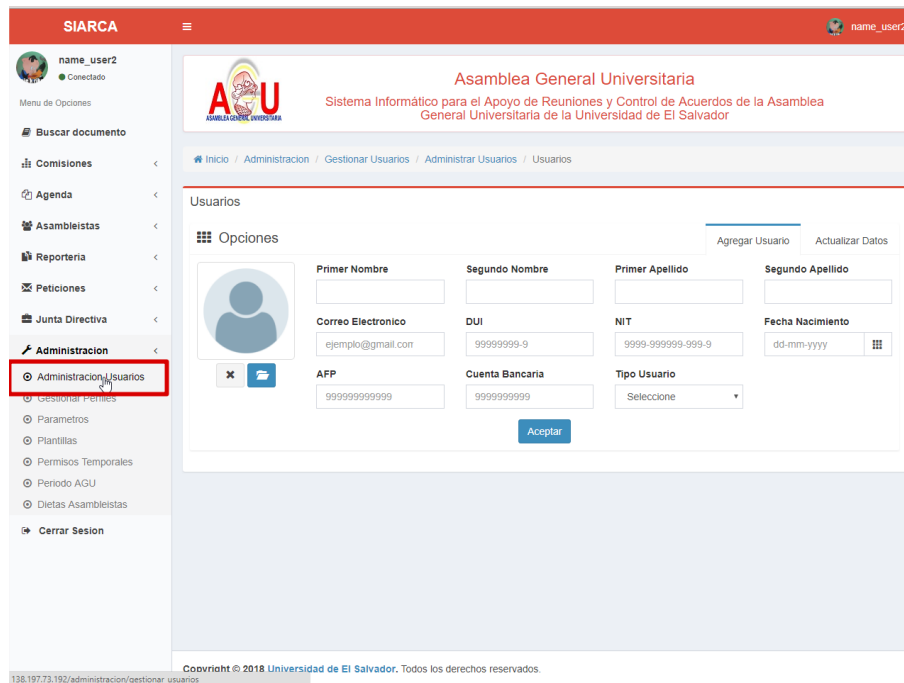


Figura 76. Vista - Registrar Usuario
Fuente: SIARCA_AGU_UES

3.1.2 Controlador

El controlador que permite la interacción entre la vista y el modelo posee las especificaciones siguientes:

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Clases\Mensaje;
6  use App\Persona;
7  use App\User;
8  use Illuminate\Http\JsonResponse;
9  use Illuminate\Http\Request;
10
11 use App\Http\Requests;
12 use Illuminate\Support\Facades\Auth;
  
```

Figura 77. Importaciones para el Registro de Usuarios
Fuente: Proyecto SIARCA, \ App\Http\Controllers\Auth\ UsuarioController.php

```

14 class UsuarioController extends Controller
15 {
16     public function mostrar_datos_usuario(){
17         $user = User::find(Auth::user()->id);
18         return view("Usuario.administrar_datos_usuario",['usuario'=>$user]);
19     }
20
21     public function actualizar_contraseña(Request $request){
22         if ($request->ajax()){
23             $user = User::find($request->id_user);
24             $user->password = bcrypt($request->password);
25             $user->save();
26             $respuesta = new stdClass();
27             $respuesta->mensaje = (new Mensaje("Exito","Contraseña actualizada con exito","success"))->
                toArray();
28             return new JsonResponse($respuesta);
29         }
30     }
31
32     public function actualizar_imagen(Request $request){
33         if ($request->ajax()){
34             $user = User::find(Auth::user()->id);
35             $persona = Persona::find($user->persona_id);
36             $disco = "../storage/fotos/";
37
38             if ($request->hasFile('img')) {
39                 $file = $request->files->get('img');
40                 $ext = $file->guessExtension();
41                 if ($ext == "jpeg" || $ext == "jpg" || $ext == "png" || $ext == "gif") {
42                     $nombreArchivo = time() . '.' . rand(1, 9999) . '.' . $file->getClientOriginalExtension
                        ();
43                     $persona->foto = $nombreArchivo;
44                     $archivo = $file->move($disco, $nombreArchivo);
45                     $persona->save();
46                 }
47             }
48
49             $respuesta = new stdClass();
50             $respuesta->mensaje = (new Mensaje("Exito","Imagen actualizada con exito","success"))->toArray
                ();
51             return new JsonResponse($respuesta);
52         }
53     }
54 }
55

```

Figura 78. Clase para el registro de usuarios

Fuente: Proyecto SIARCA, \ App\Http\Controllers\Auth\ UsuarioController.php

Capítulo IV Pruebas

4.1 Trazabilidad de casos de pruebas – requisitos²⁸

A continuación, se muestra el alcance de los casos de prueba, en relación al listado de requerimientos contemplados para el SIARCA:

Tabla 44. Matriz de Trazabilidad de casos de pruebas
Fuente: (Junta de Andalucía, 2018)

		Requerimiento																																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
Caso de prueba	PI001		X																															X	X										
	PI002	X	X	X	X	X	X	X	X	X																		X	X	X	X	X									X				
	PI003			X		X											X	X													X														
	PI004										X	X								X	X	X	X	X	X	X															X	X	X	X	
	PI005																					X						X	X																
	PI006	Pruebas de configuración del sistema, el cual se constata con el manual de instalación y desinstalación.																																											
INTEGRALES	Las realizadas por cada uno de los integrantes en la etapa de desarrollo y de forma individual.																																												

Para profundizar más sobre las pruebas de Software, consultar CD.



Véase en documento “Manual de Usuario.pdf”, adjunto en CD:
/Manuales/Plan de Pruebas-SIARCA.pdf

²⁸ Retomado de Plan de Pruebas-SIARCA

Capítulo V

Documentación

5.1 Manual de Usuario

Su objeto, es instruir a los usuarios del Sistema informático para el apoyo de reuniones y control de acuerdo de la Asamblea General Universitaria de la Universidad de el Salvador, para la realización exitosa de todas las operaciones que el SIARCA permite.



*Véase en documento "Manual de Usuario.pdf", adjunto en CD:
/Manuales/Manual de Usuario.pdf*

5.2 Manual de Instalación y desinstalación

Su objeto, es instruir al personal técnico que se encuentre a cargo de la administración del Sistema informático para el apoyo de reuniones y control de acuerdo de la Asamblea General Universitaria de la Universidad de el Salvador, para dar inicio a las actividades involucradas con la puesta en producción del sistema antes mencionado.



*Véase en documento "Manual de Instalación.pdf", adjunto en CD:
/Manuales/Manual de instalación. Pdf*

5.3 Manual Técnico

Su objeto, es instruir al personal técnico que se encuentre a cargo de la administración del Sistema informático para el apoyo de reuniones y control de acuerdo de la Asamblea General Universitaria de la Universidad de el Salvador, para una instalación exitosa del Sistema informático.



*Véase en documento "Manual Técnico.pdf", sección 7.1.1., adjunto en CD:
/Manuales/Manual Técnico.pdf*

En aspectos generales, los manuales contemplan las siguientes especificaciones:

Tabla 45. Especificación de manuales

Fuente: Elaboración propia

DOCUMENTO	DESCRIPCIÓN
MANUAL DE USUARIO	Dirigido a Usuario del negocio Documento que describe como se debe usar el SIARCA y los requerimientos para su uso, cuyo ejemplar formado por 95 páginas se adjunta en CD
MANUAL TÉCNICO	Dirigido a Administrador del sistema Documento que describe los elementos considerado para el desarrollo del Sistema Informáticos, así como aspectos que permitan la realización de mejoras o actualizaciones al SIARCA, cuyo ejemplar formado por 73 páginas se adjunta en CD.
MANUAL DE INSTALACIÓN	Dirigido al Administrador del Sistema. Documento que describe lo necesario para la instalación del Sistema informático en ambiente de producción, cuyo ejemplar formado por 18 páginas se adjunta en CD.

Capítulo VI

Plan de implementación

Introducción

Siendo el Sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador, abreviado SIARCA_AGU_UES, el primer sistema informático del tipo administrativo que posee la unidad de negocio licenciada, a continuación se presenta la información necesaria para la puesta en marcha de dicho Sistema informático; por lo que es necesario contemplar desde siglas, definiciones y abreviaturas, hasta recursos y plan de capacitación para que los usuarios del SIARCA, puedan desempeñar sus actividades con normalidad.

Propósito

El sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador, dentro de sus principales propósitos tiene:

- ✓ Colaborar en la generación oportuna de dietas para mejorar la ejecución presupuestaria de la unidad de negocio a implementarse.
- ✓ Automatizar los procedimientos llevados a cabo por los asambleístas y empleados de la AGU, permitiendo la reducción de tiempo en la realización de los mismos, así como la reducción de los recursos relacionados a las convocatorias.
- ✓ Contribuir a la transparencia del trabajo realizado por la Asamblea General Universitaria, permitiendo un seguimiento a las peticiones por parte de las partes interesadas, así como la consulta de documentos generados por el organismo.

Alcance

El documento actual se encuentra dirigido a los miembros de la Junta directiva de la Asamblea General Universitaria, de la Universidad de el Salvador, al Administrador del sistema informático y al equipo delegado para la puesta en marcha del SIARCA_AGU_UES, quiénes en conjunto, posean conocimientos de la unidad de negocios (políticas, reglas procesos, etc), así como, poseer conocimientos técnicos (Administración de servidores, desarrollo de sistemas informático, pruebas de SW y capacitación).

Definiciones, siglas y abreviaturas

Asambleísta: Es las personas electas democráticamente para representar a una Facultad, según el sector al que pertenezca.

AGU: Asamblea General Universitaria

Base de Datos: Un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Dieta: Estipendio denominado dieta por el cumplimiento de sus atribuciones

JD: Junta Directiva de la Asamblea General Universitaria.

SIARCA_AGU_UES: Sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador.

SW: Software, conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

UES: Universidad de El Salvador

Referencias

- Grupo 09. (2011). *Facultad de ingeniería y Arquitectura de la Universidad de Uruguay*.
Obtenido de Facultad de ingeniería y Arquitectura de la Universidad de Uruguay:
<https://www.fing.edu.uy/inco/cursos/ingsoft/pis/memoria/dvd02/experiencia2011/material/grupo9/implantacion/construccion/iter2/IPPLAG09V1.0.pdf>
- TBS115-G09. (2017). *Manual de instalación y desinstalación-SIARCA_AGU_UES*.
Universitaria.
- TBS115-G09. (2017). *Manual de Usuario-SIARCA_AGU_UES*. Universitaria.
- TBS115-G09. (2017). *Manual Técnico-SIARCA_AGU_UES*. Universitaria.
- Trabajo de Grado. (2017). *Sistema Informático para la Admisnitración del Proceso de Evaluación Docente de la Facultad de Ingeniería y Arquitectura de la Universidad de El Salvador*. Universitaria.

Visión general

En general, el presente plan se encuentra dividido en 3 secciones principales, las cuales hacer referencia a la Planificación de la implementación, el recurso requerido y la capacitación necesaria para el uso del Sistema informático.

Planificación de la Implantación

Las actividades involucradas para para la puesta en marcha del SIARCA_AGU_UES, contempla 5 macro categorías, las cuales se describen en la ilustración siguiente:

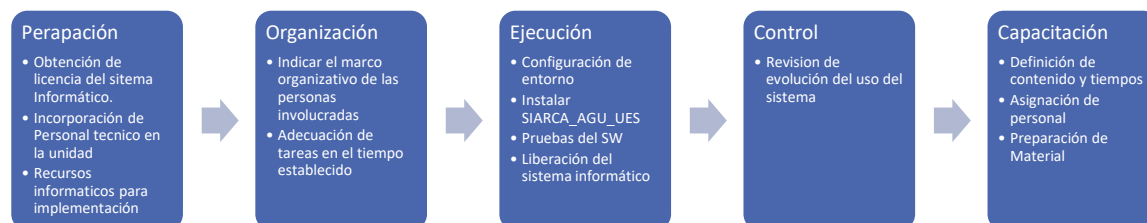


Ilustración 1. Secuencia de implementación

A. Preparación:

- La obtención de licencia consiste en solicitar de manera formal la donación del sistema informático a la Facultad de Ingeniería y Arquitectura, tal como lo establece el manual técnico en la especificación de requerimientos legales.

- La incorporación de personal, se refiere a la adhesión de un Administrador²⁹ de Sistema a la Asamblea general Universitaria, el cual será el responsable de los aspectos técnicos del SIARCA_AGU_UES
- Los recursos informáticos para la implementación, es una actividad que consiste en verificar si se cuenta con el software y hardware necesario para la puesta en marcha, en este espacio, será necesario la solicitud del espacio de producción a la dirección de tecnologías de la Universidad de El Salvador.

B. Organización

- Indicar el marco organizativo, consiste en establecer una estructura organizativa que se encargará de la implantación del Sistema informático.
- La adecuación de tareas según corresponda, hace referencia a la asignación de responsabilidades por participante según el tiempo establecido para las mismas.

C. Ejecución

- Configuración de entorno: son todas las actividades necesarias para la instalación del sistema informático, que involucra tareas de instalación de software.
- Instalar SIARCA_AGU_UES: Se refiere a la instalación en un ambiente de prueba para asegurarse de la liberación sin problemas del Sistema informático.
- Pruebas del SW: consiste en repetir las pruebas indicadas por los desarrolladores en el documento de pruebas, para corroborar la veracidad de las mismas e identificar posibles errores a solventar.
- Liberación del sistema informático, consiste en un hito, que libera una versión en ambiente de desarrollo para la realización de la capacitación y posteriormente, migrar el sistema informático a producción.

²⁹ Para mayor detalle del perfil, consultar manual técnico.

D. Control³⁰

- Revisión de evolución en uso del sistema, verificando una vez se vayan realizando las pruebas, cuyo objetivo es validar el nivel de integración del sistema informático.

E. Capacitación

- Definir contenido y tiempos: se debe definir el plan de capacitación a cubrir.
- Asignación de personal: hace referencia a delegación de personas encargadas de impartir la capacitación
- Preparación de Materia: contempla la segmentación de material según tipo de usuario y Reproducción, esto a partir del manual de usuario.

Responsabilidades

El equipo a cargo de la implementación del SIARCA_AGU_UES es el descrito en la ilustración siguiente:

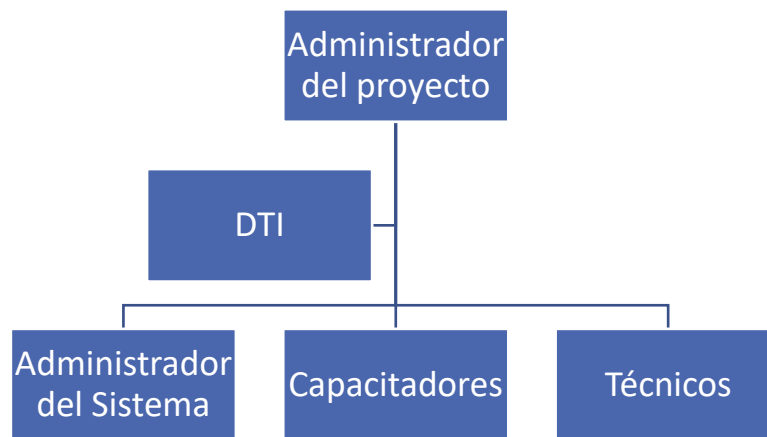


Ilustración 2. Responsables de implementación del Sistema informático
Fuente: Elaboración propia

Donde:

- a) Administrador de Proyecto (AP), será el presidente/a de la Asamblea General Universitaria.

³⁰ La medición de resultados puede realizarse por medio de los índices de control especificados en el anexo denominado: "Índice de control"

- b) DTI (D): Dirección de Tecnologías de la Universidad de El Salvador, como unidad de apoyo para la selección o asignación de personal técnico a cargo de la Asamblea General Universitaria.
- c) Administrador del Sistema(AS): personal a cargo de la administración del SIARCA_AGU_UES
- d) Capacitadores(C): Personal designando para impartir capacitaciones del uso de sistemas a usuarios de negocio.
- e) Técnicos(T): Personal asignado para colaborar con el Administrador del Sistema para la implantación del SIARCA_AGU_UES

Las responsabilidades se definen en la siguiente matriz:

Tabla 46. Matriz de Asignación de responsabilidades
Fuente: Elaboración propia

ACTIVIDAD	RESPONSABLE				
	AP	D	AS	C	T
OBTENCIÓN DE LICENCIA DE USO	X				
CONTRATACIÓN DE ADMINISTRADOR DEL SISTEMA	X	X			
VERIFICACIÓN DEL HW Y SW PARA IMPLEMENTACIÓN DEL SIARCA			X		
SOLICITAR ESPACIO FÍSICO A DTI	X				
CONFIGURACIÓN DE ENTORNOS			X		X
IMPLEMENTACIÓN DE SIARCA_AGU_UES			x		
PRUEBAS DEL SISTEMA INFORMÁTICO			X		X
SOLUCIÓN DE PROBLEMAS IDENTIFICADOS			X		X
LIBERAR SIARCA EN AMBIENTE DE DESARROLLO			X		X
PRESENTACIÓN DE SISTEMA INFORMÁTICO FUNCIONANDO			X		
MIGRAR SIARCA AL AMBIENTE DE PRODUCCIÓN			X		
MONITOREAR AVANCE DE SAIRCA			X		
MONITOREAR ACEPTACIÓN DEL SIARCA	X				
CONTRATAR CAPACITADORES	X		X		
ELABORAR PLAN DE CAPACITACIÓN				X	
PREPARACIÓN DE MATERIAL PARA CAPACITACIÓN					
REPRODUCCIÓN DE MATERIAL PARA CAPACITACIÓN	X				

Cronograma

A continuación, se muestra el cronograma de actividades sugerido para la implementación exitosa del Sistema informático para el apoyo de reuniones y control de acuerdo de la Asamblea General Universitaria.

Tabla 47. Cronograma del plan de implementación
Fuente: Elaboración Propia

NO	TAREA	DURACIÓN(DÍAS)	DESCRIPCIÓN	PREDECESORA	RECURSOS
1	Obtención de licencia de uso	10	Trámite para adquirir Software como donación	-	-
2	Reclutamiento de personal	10	Actividades encaminadas para la solicitud de apoyo con personal a la DTI	-	Papelería
3	Solicitar Administrador de sistema a DTI	5	-	-	-
4	Solicitar Personal DTI	5	-	-	-
5	Gestionar Capacitadores	5	-	-	-
6	Solicitar espacio físico a DTI	5	Conciste en lo relacionado a solicitar el alojamiento del	3	-

NO	TAREA	DURACIÓN(DÍAS)	DESCRIPCIÓN	PREDECESORA	RECURSOS
			Sistema informático en los Servidores de la UES		
7	Verificación del HW y SW para implementación del SIARCA	5	Realizar check list del hardware existente y si este cumple o no con lo establecido, para el funcionamiento del SIARCA, caso contrario, se hará una lista de equipo informático a incorporar.	2	1 Administrador de Sistema 2 Analista/Progra Papelería
8	Configuración de entornos	4	Consiste en instalar los componentes que necesita el SIARCA	7	1 Administrador de Sistema 1 Analista/Progra. Papelería
9	Liberar SIARCA en ambiente de desarrollo	0	Dejar establecido una versión del Sistema para su monitoreo y realización de capacitación	8	Administrador de Sistema

NO	TAREA	DURACIÓN(DÍAS)	DESCRIPCIÓN	PREDECESORA	RECURSOS
10	Pruebas del Sistema informático	5	Valida que se haya realizado de forma correcta la instalación del sistema informático	9	2 Analista/Progra. Papelería
11	Solución de problemas identificados	4	Actividades de desarrollo para la solución de conflictos generados o determinados con las pruebas	-	1 Administrador de sistema Papelería
12	Presentación del sistema informático funcionando	0	Se debe mostrar al administrador del proyecto, que una versión del sistema informático ha sido instalada con éxito.	11	Administrador de Sistema
13	Migrar SIARCA al ambiente de producción	4	consiste en replicar lo tenido en ambiente de desarrollo para comenzar a utilizar el sistema informático	12	2 Analista/Progra.
14	Monitorear avance de SIARCA	6	Consiste en la realización de pruebas para validar que la instalación en el ambiente de producción haya sido exitosa.		Administrador de Sistema

NO	TAREA	DURACIÓN(DÍAS)	DESCRIPCIÓN	PREDECESORA	RECURSOS
15	Elaborar plan de capacitación	5	consiste en definir el temario a cubrir de acuerdo al rol que desempeñe un usuario	5	Capacitador
16	Preparación de material para capacitación	4	Preparar el material didáctico utilizar, el cual será enviado de forma digital.	15	Capacitador
17	Reproducción de material para capacitación	2	Tarea que corresponde a la Asamblea General Universitaria, la cual consiste en proporcionar recursos para la realización de las capacitaciones.	16	Papelería
18	Capacitación	13	Desarrollo de capacitaciones con el material y plan de capacitación	12	
19	Empleados	3	-		Capacitador
20	Junta Directiva	3	-		Capacitador
21	Coordinadores de comisión	3	-	20	Capacitador
22	Asambleístas	1	-	21	Capacitador

NO	TAREA	DURACIÓN(DÍAS)	DESCRIPCIÓN	PREDECESORA	RECURSOS
23	Monitorear aceptación SIARCA	7	Consiste en la observación directa a los usuarios del SAIRCA, para evaluar su nivel de aceptación.		Administrador de Sistema
24	Implantación SIARCA_AGU_UES	0	Darán inicio a las actividades de los usuarios al interior del SIARCA, desde un ambiente de producción.	18	Administrador de Sistema

Recursos

Los recursos necesarios para la puesta en marcha del SIARCA_AGU_UES, se retoman de los manuales técnicos, usuario, instalación y la sección 2.2 del presente manual, razón por la cual, a continuación se listan los recursos con sus respectivas referencias.

Hardware

Tabla 48. Especificaciones de Hardware del plan de implementación
Fuente: Elaboración Propia.

AMBIENTE	ESTACIÓN	REFERENCIA
DESARROLLO	Servidor	Manual Técnico, Sección 7.1.1: Requisitos técnicos
	Trabajo	Manual Técnico, Sección 7.1.1: Requisitos técnicos
PRODUCCIÓN	Servidor	Manual de Instalación, Sección: Recurso Hardware
	Trabajo	Manual Técnico, Sección 7.1.1: Requisitos

A responsabilidad de la Asamblea General Universitaria

El sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria, de la Universidad de El Salvador, será entregado a la Escuela de Sistemas Informáticos de la Facultad de Ingeniería y Arquitectura de la UES, a espera de ser solicitado como donación por la AGU, esto por reglamentación de trabajos de grado de la Universidad de EL Salvador.

El software solicitado contendrá en un CD, un ejemplar del trabajo de grado, el proyecto desarrollado, así como el manual de usuario, técnico, instalación y desinstalación del SIARCA_AGU_UES.

Software de apoyo

Tabla 49. Software de apoyo para el plan de implementación
Fuente: Elaboración propia.

AMBIENTE	ESTACIÓN	REFERENCIA
DESARROLLO/PRODUCCIÓN	Servidor	Manual Técnico, Sección 5: Fundamentos de Tecnología utilizadas. Sección 7.1.1: Requisitos técnicos.
	Trabajo	Manual de Usuario Sección: Requisitos de uso.

Documentación de apoyo

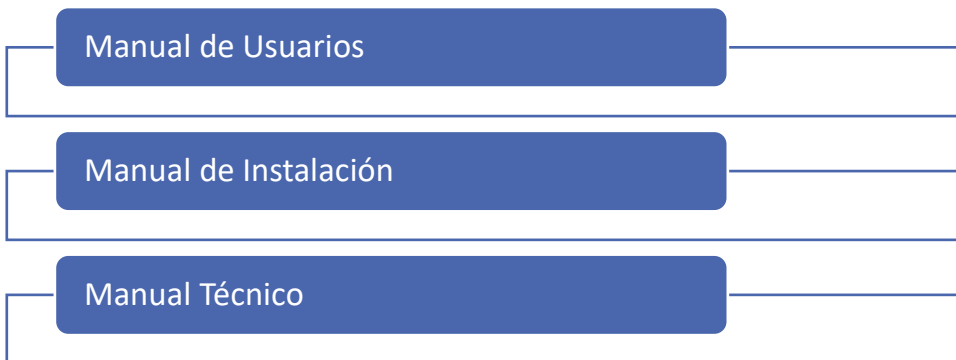


Ilustración 3. Documentación de apoyo al plan de implementación
Fuente: Elaboración propia.

Personal de apoyo

Además del administrador de Sistemas que debe incorporarse a la Asamblea General Universitaria, será necesario el apoyo de dos analistas programadores de la DTI, para colaborar en la implementación del Sistema informático.

Dichos elementos, deben poseer conocimientos en las tecnologías requeridas por Laravel V5.4, tal como se especifica en el manual técnico.

Capacitación

Para la capacitación de los usuarios de negocio se debe disponer de 2 capacitadores, que brinden jornadas de máximo 4 horas diarias, para un máximo de 3 días, tal como se mostró en el cronograma del presente plan, así mismo se necesita un plan de capacitación, el cual será creado por el capacitador, sin embargo, en la tabla siguiente se detallan los contenidos mínimos a contemplar por usuario y los tiempos sugeridos.

Tabla 50. Plan de capacitación
Fuente: Elaboración propia

USUARIO	CONTENIDO	DURACIÓN
EMPLEADOS	Propósito del Sistema informático y Generalidades	60 min
	Reconocimiento del entorno.	40min
	Ingreso al Sistema	30min
	¿A qué funciones tienen acceso?	60 min
	Desempeñando mis funciones paso a paso	200
	Ejercicios prácticos	180
	Preguntas y respuestas.	120 min
	TOTAL DE HORAS	
JUNTA DIRECTIVA	Propósito del Sistema informático	60 min
	Reconocimiento del entorno.	40min
	Ingreso al Sistema	30min
	¿A qué funciones tienen acceso?	60 min
	Desempeñando mis funciones paso a paso	200

USUARIO	CONTENIDO	DURACIÓN
	Ejercicios prácticos	180
	Preguntas y respuestas.	120 min
	TOTAL DE HORAS	11 horas con 30 min
COORDINADORES DE COMISIÓN	Propósito del Sistema informático	60 min
	Reconocimiento del entorno.	40min
	Ingreso al Sistema	30min
	¿A qué funciones tienen acceso?	60 min
	Desempeñando mis funciones paso a paso	200
	Ejercicios prácticos	180
	Preguntas y respuestas.	120 min
	TOTAL DE HORAS	11hrs con 30 min
ASAMBLEÍSTA	Propósito del Sistema informático	30 min
	Reconocimiento del entorno.	25 min
	Ingreso al Sistema	25 min
	¿A qué funciones tienen acceso?	30 min
	Desempeñando mis funciones paso a paso	60
	Ejercicios prácticos	40
	Preguntas y respuestas.	30
TOTAL DE HORAS		4 hrs

Finalmente y no menos importante, se presenta una estimación del costo que representaría la implantación del sistema informático.

a) Costo por mano de obra

ESTADO DE LOS RECURSOS

Resta trabajo para todos los recursos de trabajo

Nombre	Comienzo	Fin	Trabajo restante
Administrador de Sistema	lun 19/3/18	lun 16/4/18	156 horas
Analista/Programador 1	lun 19/3/18	mié 11/4/18	72 horas
Analista/Programador 2	lun 19/3/18	mié 11/4/18	72 horas
Capacitador 1	lun 12/3/18	vie 13/4/18	44 horas
Capacitador 2	lun 19/3/18	lun 16/4/18	32 horas

Ilustración 4. Cantidad de horas hombre requerida

Fuente: Project 2013, Estimación de recursos.

Donde los costos totales generados:

Tabla 51. Costo de mano de obra para implementación
Elaboración propia.

CARGO	COSTO POR HORA	CANTIDAD DE HORAS	TOTAL GENERADO
AS	\$5.00	156	\$780.00
AP1	\$3.00	72	\$216.00
AP2	\$3.00	72	\$216.00
C1	\$5.00	44	\$ 220.00
C2	\$5.00	32	\$160.00
TOTAL			\$1592.00

b) Costo por Papelería

Adicional a la mano de obra, se considera el costo por papelería, bajo el supuesto de que las guías se enviarán por medio digital y no será necesaria su reproducción.

Tabla 52. Costo de papelería para capacitación
Fuente: Elaboración propia.

RECURSO	TRABAJO	C/U	TOTAL
PAPEL	641	\$0.02	\$12.82
LAPICERO	204	\$0.20	\$40.80
FOLDER	208	\$0.10	\$20.8
LÁPIZ	202	\$0.15	\$30.30
TOTAL			\$104.72

Por lo que el costo de implementación sin la adquisición de Hardware sería:
\$1696.62.³¹

³¹ Considérese que el pago del personal asignado por DTI, ya posee un salario propio y no representaría una erogación para la AGU.

Anexos (Plan de implementación)

A. Índices de control

Se utilizan para medir las actividades en un plan de implantación, siendo los indicadores más utilizados:

- i. **Índice de Actividades Programadas (IAEPE):** Para determinar grado de avance

$$\text{IAEPE} = \frac{\Sigma \text{Duración de las Actividades programadas ya Ejecutada}}{\Sigma \text{Duración de todas las Actividades programadas}}$$

Finalidad:

- Si el resultado del índice es menor que 1 se determina que el grado de avance del proceso de implementación, se encuentra en estado aceptable, aunque debe evaluarse el número de actividades programadas ejecutadas contra el número total de actividades.
- Si el resultado es mayor que 1 deben tomarse medidas correctivas.

Ilustración 5. Cálculo IAEPE

Fuente: (Sistema Informático para la administración del proceso de evaluación docente de la Facultad de Ingeniería y Arquitectura de la Universidad de El Salvador, 2017)

- ii. **Índice de Duración d Actividades (IDA):** Para determinar grado de desviación entre tiempo real versus el esperado

$$\text{IDA} = \frac{\text{Duración real de la actividad}}{\text{Tiempo programado para la actividad}}$$

Finalidad:

- Si el resultado del índice es menor que 1, se determina que el grado de avance de dicha actividad se encuentra en estado aceptable
- Si el resultado es mayor que 1 deben tomarse medidas correctivas.

Ilustración 6. Cálculo de IDA

Fuente: (Sistema Informático para la administración del proceso de evaluación docente de la Facultad de Ingeniería y Arquitectura de la Universidad de El Salvador, 2017)

iii. **Índice de Actividades Programadas Retrasadas(IAPR):** Para determinar grado de retraso a partir de fecha de inicio

$$IAPR = \frac{\Sigma \text{Tiempo de retraso de las actividades}}{\Sigma \text{Duración de todas las Actividades programadas}}$$

Finalidad:

- Si el resultado es cercano a cero, se determina que una actividad se encuentra en niveles aceptables.
- Si el resultado es cercano a 1 o mayor que 1, se deben tomar medidas correctivas.

Ilustración 7. Cálculo IAPR

Fuente: (Sistema Informático para la admisnitración del proceso de evaluación docente de la Facultad de Ingeniería y Arquitectura de la Universidad de El Salvador, 2017)

Conclusiones.

- 1) Los alcances del sistema informático para el apoyo de reuniones y control de acuerdo de la Asamblea General Universitaria contemplan únicamente 5 módulos, referentes a la administración de usuarios y permisos, gestión de reuniones, seguimientos de peticiones, control de documentos generados por la AGU y reportes generados.
- 2) La principal limitante con la que se contó durante el desarrollo del proyecto se encontraba relacionada a los posibles cambios de la legislación universitaria que rigen el actuar y trabajo de la Asamblea General Universitaria.
- 3) Del análisis de la situación actual se confirma el apoyo de la nueva gestión de la Asamblea General Universitaria para la consecución del sistema informático propuesto, además, se reconoce al SIARCA, como el primer sistema informático de apoyo administrativo financiero de la AGU.
- 4) El desarrollo de la aplicación inició con un total de 40 requerimientos³², incluidos en ellos los reportes solicitados por la Asamblea General Universitaria.
- 5) Se trabajaron un total de 23 Casos de usos con sus respectivos diagramas de Secuencia, para dar solución a las necesidades manifestadas por los miembros de la Asamblea General Universitaria, sin embargo, fue necesaria la incorporación de más funcionalidades para dar integridad al sistema informático y una mejor funcionalidad.
- 6) En etapa de diseño, se establecieron estándares para la Base de datos, código fuente, pantalla y reportes, congruentes a lo recomendado por el framework utilizado en desarrollo.
- 7) Las herramientas de desarrollo se adoptan bajo criterios como: compatibilidad con herramientas utilizadas al interior de la UES y experiencia de uso del equipo desarrollador.
- 8) Conjunto al Software, se hará entrega de una copia digital de la documentación siguiente:
 - a) Tomo del proyecto

³² Ver cumplimiento de ellos en anexo denominado: "Mapa de navegación del sitio"

- b) Manual de Usuario
 - c) Manual Técnico
 - d) Manual de Instalación y Desinstalación
 - e) Plan de Pruebas
 - f) Plan de Implementación
- 9) Se presenta un plan de implementación para puesta en marcha del sistema informático, el cual contempla el recurso humano, legal y tecnológico necesario para el uso del SIARCA.
- 10) Es fundamental mantener un contacto directo con la unidad de negocio a la que se le diseña un software a la medida, ya que de ellos proviene toda la información necesaria para el cumplimiento de sus necesidades, por lo cual es necesario, hacer un buen uso de los tiempos en cada reunión y validar todo lo que se vaya construyendo para evitar problemas con la entrega del producto final.
- 11) El equipo informático que actualmente posee la AGU, cumple con las especificaciones mínimas para el uso del sistema informático, no obstante, es necesario la realización de gestiones para el espacio físico donde se alojará el SIARCA_AGU_UES.

Recomendaciones

- 1) Para la comprensión de las políticas del negocio, se recomienda leer los documentos de la legislación universitaria citadas en el presente trabajo
- 2) Para futuras modificaciones, considerar las variaciones que la legislación universitaria haya sufrido, a fin de generar propuestas consistentes, que ayuden en la agilización de procedimientos realizados al interior del organismo y estos a su vez se vean reflejados en la solución informática.
- 3) La adquisición de nueva tecnología para poder contemplar nuevos métodos en el control de asistencia.
- 4) Para la obtención de Licencia de uso, el presidente o presidenta de la AGU en funciones, debe realizar la solicitud³³ de Sistema informático a la Facultad de Ingeniería y Arquitectura por medio de la Dirección de la Escuela de Ingeniería de Sistemas Informáticos.
- 5) Para la administración y puesta en marcha del sistema:
 - a) Solicitar la asignación de espacio de alojamiento para el sistema informático en los Servidores de la Universidad de El Salvador a cargo de la DTI.
 - b) Solicitar la asignación de al menos una persona por parte de la Dirección de Tecnologías de la UES para la implementación y monitoreo del SIARCA_AGU_UES.
 - c) Poner en marcha el plan de capacitación para los usuarios del Sistema informático.
 - d) Dar de baja el sitio actualmente en uso denominado NotiAGU y dar paso a la difusión del nuevo medio de difusión de acuerdos de la AGU

³³ Entregada en formato digital por parte de los miembros del trabajo de grado a la Asamblea General Universitaria; ver en anexos: "Formato de solicitud de uso licencia".

Bibliografía.

1. ALAVA, N. (2015 de Junio de 2015). *Ingeniería en Software*. Obtenido de Ingeniería en Software: <https://ingenieriaensofwarenathalyalava.wordpress.com/2015/06/01/diagramas-de-casos-de-uso/>
2. ALEJANDRO EMILIO, B. A., SANDRA GISELLE, C. S., ANA GUADALUPE, C. C., & BORIS IVÁN, L. L. (2017). *SISTEMA INFORMATICO PARA LA ADMINISTRACION DEL PROCESO DE EVALUACION DOCENTE DE LA FACULTAD DE INGENIERÍA Y ARQUITECTURA DE LA UNIVERSIDAD DE EL SALVADOR*. Universitaria.
3. Bbits. (2016 de Junio de 2016). *BBITS*. Obtenido de BBITS: <https://borrowbits.com/2016/06/4-razones-para-usar-git/>
4. Beltrán Hernández, J. C., Criollo Hernández, W. C., & Noyola Sánchez, C. A. (2016). *ETAPA II: PROYECTO- SISTEMA INFORMÁTICO PARA LA ASIGNACIÓN DE DIETAS*. Universitaria: Administración de Proyectos Informáticos.
5. BELTRAN SERRANO , J. A., SANCHEZ CASTRO, M. A., LANDAVERDE GARCIA, W. E., BOLAÑOS CANJURA, R. M., & MIRANDA RAMIREZ, R. M. (2017). *SISTEMA INFORMÁTICO PARA LA ADMINISTRACIÓN DE BECAS DE POSTGRADO EN LA UNIVERSIDAD DE EL SALVADOR*. Universitaria.
6. Beltran Serrano, J. A., Sanchez Castro, M. A., Landaverde Garcia, W. E., Bolaños Canjura, R. M., & Miranda Ramirez, R. M. (s.f.). *Sistema Informatico para la Administracion de becas de postgrado en la Universidad de El Salvador*.
7. Coleman, A. (s.f.). *self-taugh coders*. Obtenido de self-taugh coders: <https://selftaughtcoders.com/from-idea-to-launch/lesson-17/laravel-5-mvc-application-in-10-minutes/>
8. *Composer*. (2017). Obtenido de getcomposer: <https://getcomposer.org/doc/00-intro.md>
9. Criollo Hernández, W. C., Donado Iraheta, J. F., López Henriquez, J. B., & Noyola Sánchez, C. A. (2017). *ANTEPROYECTO-SIARCA_AGU_UES*. Universitaria.
10. Criollo Hernández, Wendy Carolina; Donado Iraheta, Jaime Fermín; Noyola Sánchez, Carlos Alberto; López Henriquez, Jonathan Benjamín. (2018). *SISTEMA INFORMATICO PARA EL APOYO DE REUNIONES Y CONTROL DE ACUERDOS DE LA ASAMBLEA GENERAL UNIVERSITARIA DE LA UNIVERSIDAD DE EL SALVADOR. SIARCA_AGU_UES*. Universitaria.
11. Goñi, F. (18 de Julio de 2016). *Cafe de Por medio*. Obtenido de CoffeDevs: <https://blog.coffeedevs.com/8-razones-para-usar-git/>
12. Grupo 09. (2011). *Facultad de ingeniería y Arquitectura de la Universidad de Uruguay*. Obtenido de Facultad de ingeniería y Arquitectura de la Universidad de Uruguay: <https://www.fing.edu.uy/inco/cursos/ingsoft/pis/memoria/dvd02/experiencia2011/material/grupo9/implantacion/construccion/iter2/IPPLAG09V1.0.pdf>
13. Infranetworking. (6 de Julio de 2015). *Guiadev*. Obtenido de Guiadev: <https://guiadev.com/10-razones-para-usar-laravel-5-como-framework-php/>

14. Infranetworking. (6 de Mayo de 2016). *Guiadev*. Obtenido de Guiadev: <https://guiadev.com/mariadb-vs-mysql-cual-debo-elegir/>
15. Ing. Carlos Ernesto García, M. S. (2009). *Guía de Preparación del Anteproyecto*. Universitaria.
16. IRIS IVETTE, H. A., CARLOS ANTONIO, M. L., DAVID ARISTIDES, O. A., & LISANDRO ANTONIO, R. C. (2013). *SISTEMA DE REGISTRO Y ARCHIVO DE LA DOCUMENTACION DE LA FACULTAD DE INGENIERIA Y ARQUITECTURA DE LA UNIVERSIDAD DE EL SALVADOR*. Universitaria.
17. JFernandez. (01 de 01 de 2012). *www.uv.mx*. Obtenido de *www.uv.mx*: <https://www.uv.mx/personal/jfernandez/files/2012/11/PruebaIntegracionEstructurada.pdf>
18. Junta de Andalucía. (02 de 2018). *Junta de Andalucía*. Obtenido de Plantilla Plan de Pruebas Funcionales : <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/462>
19. MARCELA YASMIN, A. P., NESTOR WILLIAN, H. O., OMAR ALFREDO, O. C., & JUAN CARLOS, S. M. (2015). *Sistema informático de apoyo a la gestión y monitoreo de servicios integrales de salud que proporciona el personal de enfermería pacientes*. Universitaria.
20. Marín, E. M. (17 de Octubre de 2014). *Edson Mollericono Marín Blog*. Obtenido de Edson Mollericono Marín Blog: <http://www.edsonmm.com/razones-para-usar-un-framework-php/>
21. Martin, M. P. (10 de diciembre de 2012). *LANCETALENT*. Obtenido de <https://www.lancetalent.com/blog/6-buenos-motivos-para-trabajar-con-php/>
22. Mendoza, D. (22 de Enero de 2016). *Daniel Mendoza*. Obtenido de Daniel Mendoza: <http://danielmlozano.com/eres-programador-de-php-debes-usar-laravel/>
23. OTWELL, T. (abril de 2017). *Laravel - The PHP Framework For Web Artisans*. Obtenido de Installation - Laravel - The PHP Framework For Web Artisans: <https://laravel.com/docs/5.4>
24. Palacios, D. (26 de Junio de 2016). *Styde*. Obtenido de Styde: <https://styde.net/porque-elegir-laravel-en-vez-de-codeigniter/>
25. *Programacion.net*. (s.f.). Obtenido de *Programacion.net*: http://programacion.net/articulo/por_que_elegir_php_143
26. *Richos-gitbooks*. (s.f.). Obtenido de <https://richos.gitbooks.io/laravel-5/content/capitulos/chapter1.html>
27. Salvador, U. d. (7 de Abril de 2008). Contenido mínimo del Anteproyecto. Universitaria.
28. *sistemasvd*. (marzo de 2017). Obtenido de Fases del Proceso de Desarrollo del Software: : https://sistemasvd.files.wordpress.com/2008/07/fases_desarrollo.jpg
29. *StarUML*. (s.f.). Obtenido de <http://staruml.io/>
30. TBS115-G09. (2017). *Manual de instalación y desinstalación-SIARCA_AGU_UES*. Universitaria.
31. TBS115-G09. (2017). *Manual de Usuario-SIARCA_AGU_UES*. Universitaria.
32. TBS115-G09. (2017). *Manual Técnico-SIARCA_AGU_UES*. Universitaria.

33. THEMES, M. N. (s.f.). *Matriz FODA*. Obtenido de <http://www.matrizfoda.com/dafo/>
34. Trabajo de Grado. (2017). *Sistema Informático para la admisnitración del proceso de evaluación docente de la Facultad de Ingeniería y Arquitectura de la Universidad de El Salvador*. Universitaria.
35. *unijimpe*. (25 de septiembre de 2007). Obtenido de <http://blog.unijimpe.net/10-razones-para-usar-php/>
36. UNIVERSIDAD DE EL SALVADOR. (2013). *Legislación Universitaria, Libro Segundo: Recopilación Académica*. Universitaria: Imprenta Universitaria.
37. UNIVERSIDAD DE EL SALVADOR. (2013). *LIBRO PRIMERO: Recopilación Adminsitrativo-financiero*. Universitaria: Editorial Universitaria.
38. UNIVERSIDAD DE EL SALVADOR. (2013). *LIBRO TERCERO: RECOPIACION DISCIPLINARIA*. Univertsitaria: Imprenta Universitaria.
39. ZOEKAT. (3 de Diciembre de 2013). *VOZIDEA*. Obtenido de VOZIDEA: <http://www.vozidea.com/que-es-mariadb-y-ventajas-frente-mysql>

Glosario.

Atestados: Documento oficial en que se da fe de un hecho.

Base de datos: es una colección de datos, la cual contiene información relevante para una organización; es descrita por medio de modelos de datos.

Casos de Uso: secuencia de acciones realizadas por un sistema en respuesta a la interacción del actor y el mismo sistema

Caso de prueba: conjunto de condiciones o variables bajo las cuáles un analista determinará si una aplicación, un sistema software (software System), o una característica de éstos es parcial o completamente satisfactoria.

Diagrama de Secuencia (DSS): Es un tipo de diagrama usado para modelar interacción entre objetos en un sistema según UML. Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso.

Framework: Un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

Hardware: Partes físicas tangibles de un sistema informático; sus componentes eléctricos, electrónicos, electromecánicos y mecánicos.¹ Cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado componen el hardware; contrariamente, el soporte lógico e intangible es el llamado software.

Hito: Acontecimiento puntual y significativo que marca un momento importante en el desarrollo de un proceso o en la vida de una persona.

Licencia GPL: o General Public License, desarrollada por la FSF o Free Software Foundation, hace referencia a que se puede instalar y usar un programa en uno o varios ordenadores, sin limitación, quedado la potestad de modificarlo y distribuirlo de forma libre.

Modelo: En base de datos, es un conjunto de herramientas conceptuales para describir sus datos, sus relaciones, su significado y restricciones que la base de datos debe obedecer.

Modelo Conceptual: Orientados a representar la percepción del mundo por medio de objetos llamados entidades y de relaciones entre estos objetos. Los modelos conceptuales, por ende, son utilizados para representar la realidad a un alto nivel de abstracción. Este tipo de modelado de datos permite construir una descripción de la realidad fácil de entender.

Modelo Lógico: El modelo lógico, es una descripción de la estructura de la base de datos en términos de las estructuras de datos que puede procesar un tipo de SGBD. Están orientados a las operaciones más que a la descripción de la realidad, el ejemplo de modelo que se ha establecido actualmente como el principal modelo de datos para las aplicaciones de procesamientos de datos, es el Modelo Relacional.

Modelo Físico: Es una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos. Por ello, el diseño físico depende del SGBD concreto y el esquema físico se expresa mediante su lenguaje de definición de datos.

Pruebas integrales: son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias y lo que prueban es que todos los elementos unitarios que componen el software, funcionan juntos correctamente probándolos en grupo. Se centra principalmente en probar la comunicación entre los componentes y sus comunicaciones ya sea hardware o software.

Pruebas unitarias: es una forma de comprobar el correcto funcionamiento de una unidad de código, sirve para asegurar que cada unidad funcione correctamente y eficientemente por separado.

Quorum: número de personas que se necesitan para que el órgano parlamentario debate ciertos asuntos y pueda tomar una decisión válida, además este concepto jurídico es de gran importancia en el ámbito político.

Requerimiento: es una capacidad o condición que el nuevo sistema informático deberá cumplir.

Software: Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

UML: Lenguaje Unificado de Modelado, es un lenguaje para especificar, visualizar, construir y documentar los artefactos de los sistemas software, así como para el modelado del negocio y otros sistemas no software

Anexos

a) Sitio Web AGU

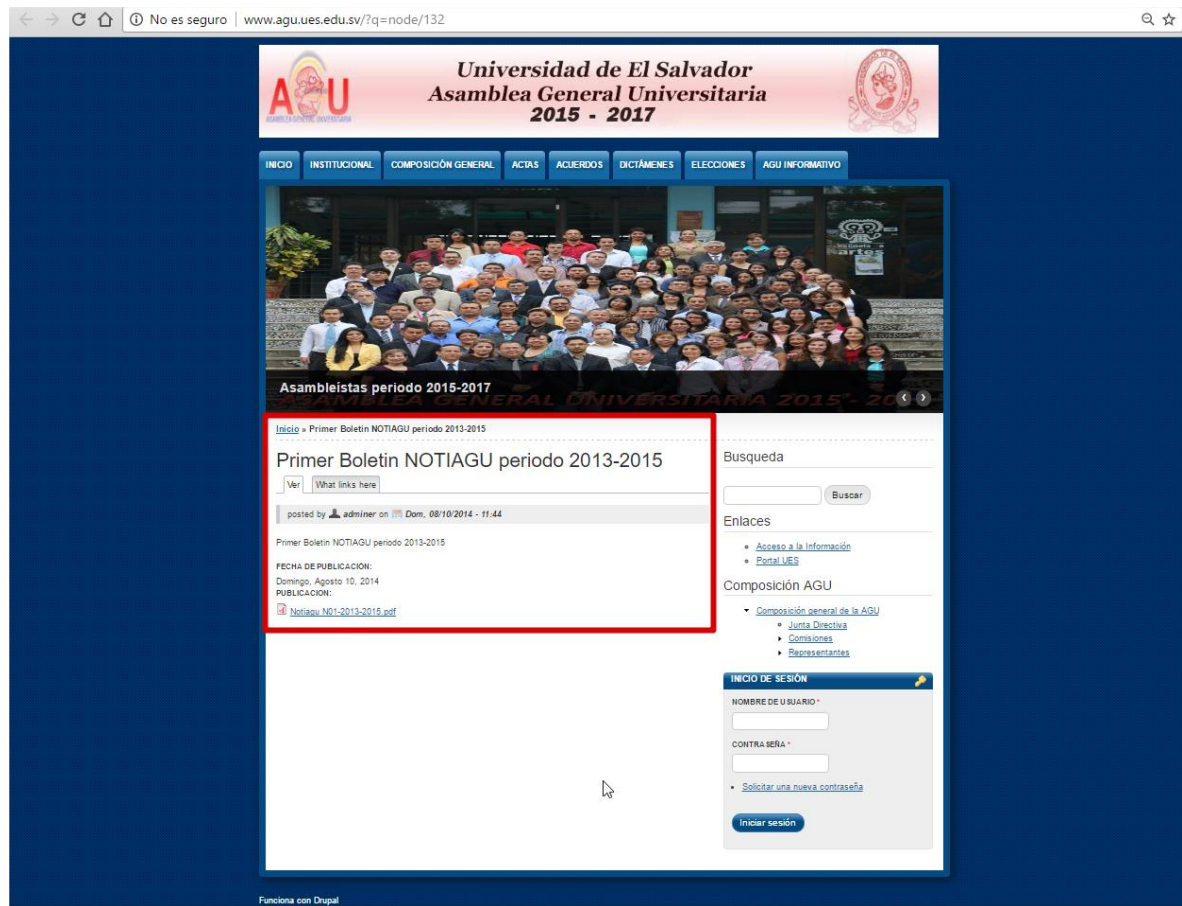




Figura 79. Sitio de información AGU
Fuente: <http://www.agu.ues.edu.sv/>

b) Acuerdo de autorización

**UNIVERSIDAD DE EL SALVADOR**
ASAMBLEA GENERAL UNIVERSITARIA



CIRCULAR

Para : Consejo Superior Universitario, Fiscalía General, Defensoría de los Derechos Universitarios, Secretaría General, Secretaría de Comunicaciones de la UES, Junta Directiva de la Asamblea General Universitaria, Auditoría Interna-UES, Fiscalía General, Oficial de Información de la UES, Decanos y Juntas Directivas de las diferentes Facultades.

De : Dra. Gloria Estela Gómez de Pérez – Secretaria.

Asunto: Transcripción de acuerdo.

Fecha : Martes 18 de marzo de 2014.

.....


Para su conocimiento y efectos legales consiguientes, transcribo a ustedes el Acuerdo N°39/2013-2015 (V), tomado en Sesión Plenaria Extraordinaria de la Asamblea General Universitaria, celebrada el viernes catorce de marzo de dos mil catorce, que literalmente dice:


V. Análisis, discusión y toma de Acuerdo sobre las Actas de Sesiones Plenarias de la Asamblea General Universitaria del período 2013-2015, solicitadas por la Oficial de Acceso a la Información Pública.

La Asamblea General Universitaria, con base en los artículos 18, 19, 81 de la Ley Orgánica; 20, 21 lit. "s" del Reglamento Interno de este Organismo, ambos de la Universidad de El Salvador por 53 votos a favor, 0 en contra y 6 abstenciones **ACUERDA:**

1. Revocar el Acuerdo N°59/2011-2013 (VIII) tomado en Sesión Plenaria Extraordinaria de la Asamblea General Universitaria, celebrada el viernes ocho de junio de dos mil doce que en lo esencial literalmente dice: **"Considerar como información reservada los audios que hubiere de las Sesiones Plenarias, Juntas Directivas y de Comisiones de la Asamblea General Universitaria de la Universidad de El Salvador".**
2. Reiterar que las Sesiones Plenarias de la Asamblea General Universitaria son públicas.
3. A partir de este día viernes catorce de marzo de dos mil catorce, las Sesiones Plenarias de la Asamblea General Universitaria, con el apoyo de la Secretaría de Comunicaciones de esta Institución, serán transmitidas por la Radio "YSUES" ubicada al Interior del Campus de la Universidad de El Salvador.
4. **Facultar a la Junta Directiva de la Asamblea General Universitaria, para que a través de su Presidente, Ing. Agr. Nelson Bernabé Granados Alvarado, realice todas las gestiones necesarias para garantizar la buena marcha y administración de este Organismo de Gobierno.**
5. **Que la Junta Directiva haga lo pertinente para mantener al día el trabajo de la Asamblea General Universitaria.**
6. Facultar a Sra. Secretaria de la Junta Directiva de este Organismo para que pronuncie:
 - a) La formulación de las Actas de Sesiones Plenarias (N° 33/2013-2015 del 31 de enero de 2014; N° 34, 35, 36 y 37/2013-2015 del 07, 14, 21 y 28 de febrero de 2014 respectivamente; y N° 38/2013-2015 del 07 de marzo de 2014) solicitadas por la Oficial de Información de la Universidad de El Salvador según las notas presentadas en fechas 10, 11 y 12 de marzo de 2014.
 - b) Que remita al Pleno de la Asamblea General Universitaria las actas citadas en el literal anterior, previo a ser firmadas con base a lo estipulado en los artículos 21 lit. "k" y 23 lit. "k" del Reglamento Interno de este Organismo, a fin que puedan ser aprobadas y que el Sr. Presidente pueda dar respuesta a lo solicitado por la Oficial de Información de la UES.

"HACIA LA LIBERTAD POR LA CULTURA"


Dra. Gloria Estela Gómez de Pérez
Secretaria
Asamblea General Universitaria



Notifíquese.

Final 25 Avenida Norte, Ciudad Universitaria, Telex. 20794, Tel. y Fax 2225-7076 Conmutador 225-1500, Ext. 4120, 4121 y 4122

Figura 80. Acuerdo para autorización de administración de recursos para la gestión de información AGU
Fuente: Archivo Asamblea General Universitaria

c) Carta compromiso

La Infrascrita Secretaria en Funciones de la Asamblea General Universitaria, de la Universidad de El Salvador del periodo 2015-2017, certifica que: En nuestros registros se encuentra el documento Original en el que se notifica formalmente a la Junta Directiva de la Asamblea General Universitaria de la realización del proyecto denominado: "Sistema Informático para el Apoyo de Reuniones y Control de Acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador, SIARCA_AGU_UES", como Trabajo de Graduación, para optar al Título de Ingeniero/a de Sistemas Informáticos, en la Facultad de Ingeniería y Arquitectura de la Universidad de El Salvador, y que literalmente dice:



Ciudad Universitaria, 10 de marzo de 2017.

Junta Directiva
Honorable Asamblea General Universitaria
Presente.

Reciba un cordial saludo deseándole éxitos en sus actividades diarias.

Sirva la presente para hacer formal la realización del proyecto denominado: **SISTEMA INFORMÁTICO PARA EL APOYO DE REUNIONES Y CONTROL DE ACUERDOS DE LA ASAMBLEA GENERAL UNIVERSITARIA DE LA UNIVERSIDAD DE EL SALVADOR, SIARCA_AGU_UES** como Trabajo de graduación, para optar al título de Ingeniero/a de Sistemas Informáticos. Donde será necesaria su colaboración así como de las personas que están a su cargo en dicho organismo y los bachilleres responsables de la creación del Sistema Informático antes mencionado.

Por lo antes expuesto, el presente escrito será garante de que durante el SIARCA_AGU_UES se encuentre en proceso, El organismo por medio de sus cargos y empleados a disposición nos brindará la información necesaria, así como el tiempo para recolección de la misma y el uso de equipo informático para la realización de pruebas del Sistema; a la vez, nosotros nos comprometemos entregar El Sistema informático funcional, listo para implementarse, así como sus respectivos manuales (Técnico, de Uso, Instalación y Desinstalación), además de un plan de implementación.

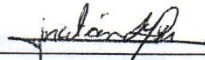
Sin más que agregar y conocidos los compromisos de las partes, sustentamos el estar de acuerdo con nuestras firmas:

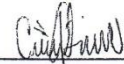
Figura 81. Carta de compromiso
Fuente: Elaboración propia

Equipo Responsable


Br. Wendy Carolina Criollo Hernández


Br. Jaime Fermín Donado Iraheta,

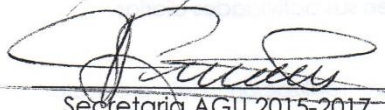

Br. Jonatán Benjamín López Henríquez


Br. Carlos Alberto Noyola Sánchez

Miembros Junta Directiva AGU


Presidente AGU 2015-2017


Vicepresidente AGU 2015-2017


Secretaria AGU 2015-2017


Segundo Vocal AGU 2015-2017

La presente copia que consta de **un folio** frente y vuelto, es conforme con su Original con el cual se confronto en la Secretaría de la Asamblea General Universitaria de la Universidad de El Salvador, y para los usos que estime la **Br. Wendy Carolina Criollo Hernández**, se extiende, firma y sella la presente en la Ciudad Universitaria a los quince días del mes de marzo de dos mil diecisiete.


Licda. Josefina Sibrian de Rodriguez
Secretaria en Funciones
Asamblea General Universitaria
Universidad de El Salvador



Figura 82. Carta compromiso, firmas sustentantes
Fuente: Elaboración propia

d) Asignación de Dietas

Tabla 53. Criterios para asignación de Dietas
Fuente: Elaboración propia, Reglamento Interno AGU.

CONDICIÓN	¿Es Asambleísta propietario?	SI	SI	SI	SI	NO	NO	NO	NO	NO	NO	NO
	¿Participó en la resolución de asuntos trascendentales?	NO	SI	NO	SI	NO	SI	NO	SI	SI	NO	NO
	¿Es suplente en carácter de propietario?	NO	NO	NO	NO	SI	SI	SI	SI	NO	NO	NO
	¿Permaneció el 80% de la sesión?	SI	NO	NO	SI	SI	NO	NO	SI	NO	NO	NO
	¿Elaboró dictamen?	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI	NO
	¿Se conoció dictamen trabajado en la sesión?	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI	NO
	¿Abandonó la sesión antes sin notificar?	NO	NO	SI	SI	NO	NO	SI	SI	NO	NO	NO
	¿Se estableció acuerdo de pago de dieta?	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI
ACCIÓN	Asignar dieta	X	X			X	X			x	x	X
	Perdida de dieta			X	X			X	X			

e) Cuestionario Análisis y Diseño

1. ¿Un asunto puede ser visto por más de una comisión?
2. ¿Es posible que el coordinador de la comisión el día de la misma puede agregar más puntos de los especificado en la convocatoria? (¿antes o durante la comisión también se puede hacer este cambio?)
3. ¿cómo modelan las dietas, debido a que los puntos trascendentales generan dieta?
4. ¿Poseen un criterio de clasificación de la documentación recibida en carácter de correspondencia? Ej. Acata, denuncia, Acuerdo de petición, otros.
5. ¿En la documentación generada por la comisión, se hace referencia al período de esta?
6. ¿Qué trato se les da a las dietas al ser vistas en agendas con puntos trascendentales?
7. ¿cómo tratan la asistencia en cuanto a dietas en comisiones y JD?
8. ¿Qué trato se les da a los dictámenes firmados por asambleísta que no asisten la sesión en el que son tocados?
9. ¿Existe relación de los puntos normales de agenda con la asignación de dietas?
10. ¿Quién es el encargado de tratar la lista de reestructuración de comisiones?
11. ¿Cuál es el criterio de selección de los puntos tomados por la JD para proponerlos al pleno?
12. ¿Poseen tiempo de respuesta para puntos específicos recibidos en la AGU?
13. ¿Se genera acta en el caso de no aprobar agenda?, ¿Qué ocurre si no se logró establecer Quorum?
14. ¿Se elabora un acata parcial de aquellas agendas que se suspenden, es decir, se dejan abiertas?

f) Plantilla Web SIARCA_AGU_UES

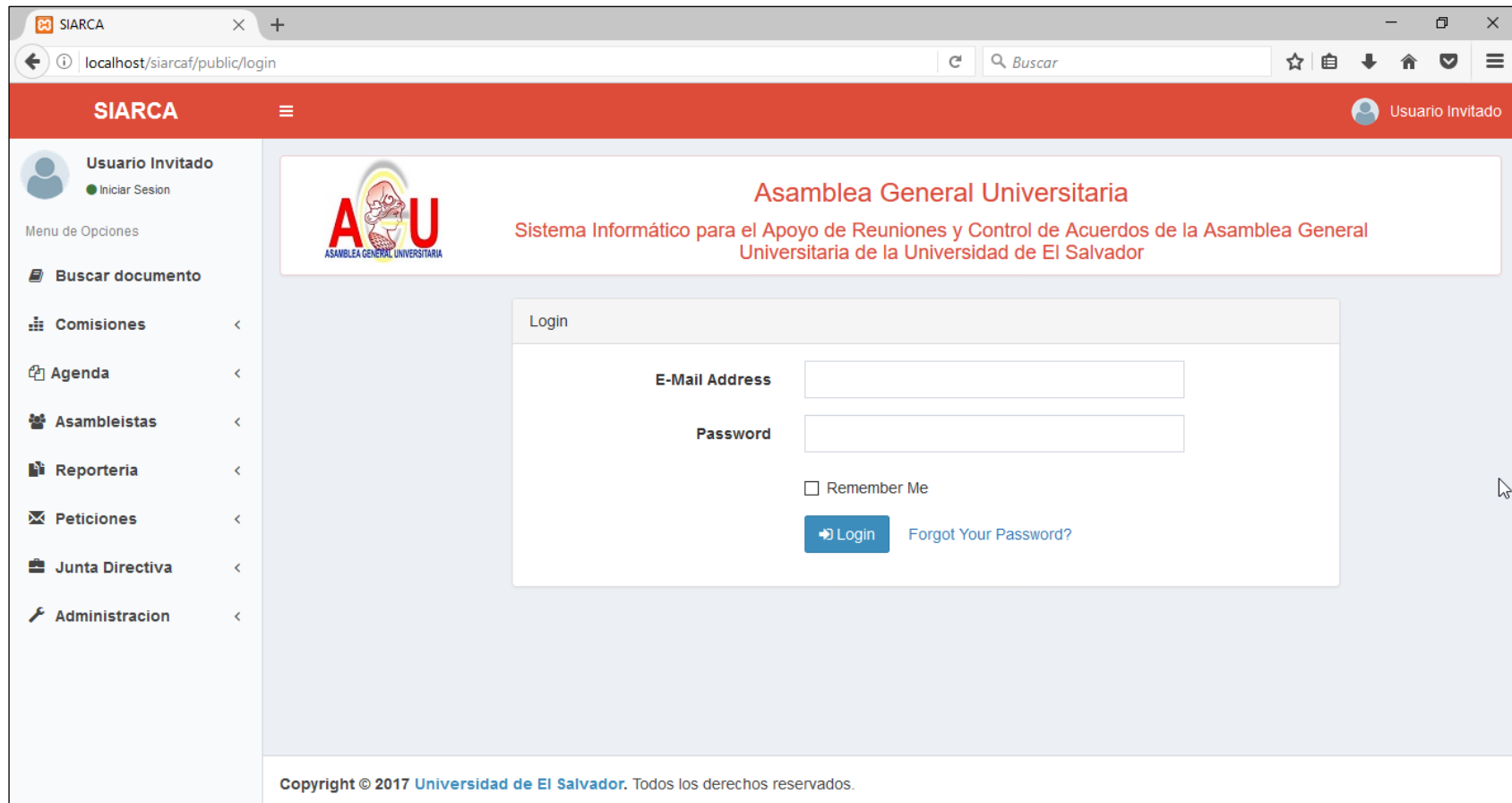


Figura 83. Vista Real de Plantilla adoptada en SIARCA
Fuente: SIARCA_AGU_UES

g) Carta de aceptación de requerimientos

Ciudad Universitaria, 04 de septiembre de 2017.

**Presidenta
Asamblea General Universitaria.
Presente.**

Considerando que los alcances del **Sistema informático para el apoyo de reuniones y control de acuerdos de la Asamblea General Universitaria de la Universidad de El Salvador**, que se abrevia **SIARCA_AGU_UES**, contempla los módulos siguientes:

1. Módulo de administración de roles, que permita, el ingreso o baja de usuarios, así como la gestión de permisos para las diferentes funcionalidades del sistema, creación de perfiles y definición de parámetros.
2. Módulo de gestión de reuniones (sesiones plenarias y comisiones) el cual permitirá: El control de sesiones plenarias, asistencias y permisos a las sesiones plenarias; administración de las agendas a tratar en las sesiones plenarias, así como, la administración del trabajo de las comisiones con tareas como: la elaboración de convocatorias a comisiones y la incorporación de asambleístas en las comisiones de la AGU.
3. Módulo que permitirá la gestión y el almacenamiento de la documentación generada por la AGU.
4. Un módulo que permitirá el control y seguimiento de las solicitudes recibidas en la Asamblea General Universitaria por parte de la comunidad Universitaria.
5. Un módulo de generación de reportes para los tres niveles organizativos estratégico, táctico y operativo asociados con la gestión de reuniones y acuerdos tomados por la Asamblea General Universitaria, permitiendo la generación en pantalla o en de forma impresa según el caso.
6. Documentación relacionada con el uso del Software, su instalación, desinstalación y modificaciones futuras.

**Figura 84. Carta de Aceptación de requerimientos
Parte 1**

Fuente: Elaboración Propia

Por lo antes expuesto, a continuación, hacemos expresas las necesidades identificadas en la unidad a su cargo, las cuales serán contempladas en la construcción del SIARCA_AGU_UES, como requerimientos únicos, los que no podrán ser ampliados o modificados posterior a su firma y en desarrollo, a menos de que sea modificados previo a la firma del presente escrito.

Los aspectos que se considerarán en la construcción del sistema son:

A. Funcionalidades de Ingreso de datos al Sistema:

- Agregar Comisión
- Incorporar Asambleístas a comisiones.
- Crear sesión plenaria (JD/AGU)
- Crear Agenda de trabajo de sesiones plenarias.
- Modificar Agendas de trabajo de sesiones plenarias
- Gestionar puntos de Agenda
- Registro de peticiones
- Crear Período
- Asignar parámetros del período
- Registro de Usuarios
- Gestionar Perfiles de Acceso
- Tomar Asistencia de Asambleístas
- Registro de permisos (Temporales/Permanentes)
- Ingresar Documentos Oficiales
- Actualización de plantillas
- Notificación de convocatoria de comisiones

B. Plantilla (Fijas y Dinámicas)

- Formato Dictamen
- Reporte de trabajo por comisión.
- Formato de Acuerdo.
- Formato de Actas
- Formato de Permisos

C. Salidas en papel o pantalla.

- Documentos Generados por AGU
- Consultar agendas vigentes
- Consultar agendas Históricas
- Listado de Asambleístas
- Listados de asambleístas por comisión
- Listados de asambleístas miembros de la Junta Directiva.

**Figura 85. Carta de Aceptación de requerimientos
Parte 2**

Fuente: Elaboración Propia

- Reporte de permisos temporales por sesión plenaria.
- Reportes de permisos permanentes por sesión plenaria.
- Reporte de asistencia de asambleístas por sesiones plenarias
- Reporte sobre bitácora de correspondencia
- Planilla de pago de Dietas por sector.
- Reporte de consolidado de renta por sector
- Constancias de rentas por Asambleístas anuales.
- Reporte de renta de Asambleístas por el Sector Docentes por Facultad.
- Listado de Asambleístas por dictámenes Emitidos
- Reporte con Consolidado de asistencias, inasistencias y permisos para un período de tiempo
- Listado de Asambleístas dados de Baja dentro de una gestión.
- Listado de Dictámenes emitidos por comisión
- Monitoreo de Petición

Finalmente, dando lectura a los requerimientos anteriores, los abajo firmantes acuerdan:

1. Dan por aceptos la propuesta de requerimientos señalados, los cuales no podrán disminuir o aumentar a partir de su firma.
2. Gestionar la información requerida para la validación de los requerimientos señalados.
3. Solicitar el apoyo de la Dirección de Tecnologías de Información de la UES (DTI), para la implementación del Sistema Informático propuesto, siendo esta una actividad ajena al equipo desarrollador del SIARCA_AGU_UES.
4. Pactar las reuniones que sean necesarias para la finalización del proyecto.
5. Los futuros cambios del sistema informático serán responsabilidad de la AGU por medio de la DTI o persona asignada para tales labores.

Firman las partes involucradas:

Josefina Silbrón de Rodríguez, Presidenta AGU

Wendy Carolina Criollo Hernández, Analista SIARCA_AGU_UES

**Figura 86. Carta de Aceptación de requerimientos
Parte 3**

Fuente: Elaboración Propia

h) Herramientas de Recolección de Información

i. Entrevista.



UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA DE SISTEMAS INFORMÁTICOS
TRABAJO DE GRADUACIÓN

ENTREVISTA ANTEPROYECTO

Objetivo: Determinar aspectos fundamentales a ser plasmados en el anteproyecto, así como aspectos que determinen antecedentes sobre Sistemas informáticos en la Unidad y los beneficios de la creación del *sistema informático para el apoyo de reuniones y control de acuerdos de la asamblea general universitaria de la universidad de el salvador*

Indicaciones: A continuación, se presentan una serie de interrogantes las cuales serán llenadas por el entrevistador; la información obtenida del presente instrumento es para uso exclusivo para el trabajo de graduación y se guardará confidencialidad de los nombres que se reflejen en ellas.

Fecha: _____

I. Clasificación del entrevistado

Nombre: _____ Edad: ____ años.

Cargo: _____

Tiempo de laborar para la Asamblea General Universitaria. _____

II. Antecedentes.

¿Conoce sobre la existencia de algún Sistema informático para el apoyo de reuniones y control de Acuerdos de la Asamblea General Universitaria? _____

Si su respuesta es sí, Describa su funcionamiento o generalidades.

III. Formulación del problema

¿Cuáles son los principales problemas que se perciben en la AGU en relación al manejo de reuniones (sesiones plenarias, sesiones de junta directiva, comisiones, etc.) y el control de acuerdos emitidos por el organismo?

¿Considera que los tiempos para la emisión de agendas, acuerdos, dictámenes y otros documentos generados por el organismo y sus comisiones son los adecuados?

Si la respuesta es no, ¿Cuáles tiempos se considerarían ideales?

IV. Beneficios esperados

¿Qué problemas creería que el SIARCA_AGU_UES podría resolver en el organismo?

¿Quiénes considera serían los principales beneficiarios con la implementación del sistema informático?

V. Sobre factibilidades

a) ECONOMICA

¿Cuántos acuerdos, dictámenes se generan en promedio en el mes?

¿Cuánto tiempo emplea en la generación de las dietas, el registro de las asistencias de los asambleístas a las sesiones plenarias, generación de la agenda a tratar en dichas sesiones, elaboraciones de actas, así como la emisión de acuerdos?

¿En dónde y de qué forma se almacenan los acuerdos, dictámenes generados y otros similares?

¿Cuánto tiempo se emplea para la búsqueda de estos últimos?

¿Cuál es el tiempo promedio invertido para la digitalización de los documentos antes mencionado?

¿Cuál sería el costo promedio mensual generado por la impresión de los acuerdos, dictámenes, reproducción de agendas, atestados y planillas de dietas?

¿Cuánto se gasta en promedio en cartuchos de tinta empleados en el mes?

ii. Encuesta



UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA DE SISTEMAS INFORMÁTICOS
TRABAJO DE GRADUACIÓN

ENCUESTA ANTEPROYECTO

Objetivo: Determinar aspectos fundamentales a ser plasmados en el anteproyecto, así como aspectos que determinen antecedentes sobre Sistemas informáticos en la Unidad y los beneficios de la creación del *sistema informático para el apoyo de reuniones y control de acuerdos de la asamblea general universitaria de la universidad de el salvador*

Indicaciones: A continuación, se presentan una serie de interrogantes las cuales deberá marcar con una **X** la que mejor se asemeje y contestar según sea el caso.

Fecha: _____

I. Identificación del encuetado

Sector al que pertenece:

Estudiantil: ___ Académico: ___ Profesional no Docente: ___

Facultad a la que representa: _____

II. Preguntas

1. ¿Considera usted que la creación de un sistema informático, que permita administrar de manera eficiente el proceso de creación y gestión de reuniones y acuerdos llevados a cabo por parte de la AGU, así como un control de las peticiones que ella recibe incluyendo la generación y cálculo de dietas, seria de utilidad para las labores diarias de sus miembros?

Sí _____ No _____

Porque:

2. ¿Cuán conveniente considera el almacenamiento de acuerdos y actas tratadas en las sesiones plenarias en formato digital como parte de la funcionalidad del sistema propuesto?

Sí _____ No _____

Porque:

3. Referente al pago de dietas a los assembleístas, ¿Considera usted, que se genera de una manera eficiente?

Sí ____ No ____

Porque:

4. Considera usted que la generación automática de un documento en el cual se presenten el nombre de los assembleístas con derecho a dieta y el monto a pagar, sería de beneficio para la AGU.

Sí ____ No ____

Porque:

5. A su criterio, ¿Se lleva un control eficiente de las peticiones recibidas por parte de la comunidad universitaria en el organismo?

Sí ____ No ____

Porque:

6. Cree que sería de utilidad contar con una opción que permita dar seguimiento a las peticiones realizada por parte de la comunidad universitaria desde su ingreso hasta su resolución dentro del Sistema informático a crear para este organismo.

Sí ____ No ____

Porque:

7. De las opciones siguientes seleccione aquellas que serían de utilidad para sus labores diarias como miembro de la AGU:

Listado de Asambleístas con derecho a dieta y la cantidad a pagar.

Listado de los puntos a tratar en las sesiones plenarias

Listado de solicitudes de peticiones a la AGU

Listado de acuerdos realizados en un periodo de tiempo

Listado de peticiones según su estado.

Listado de dictámenes efectuados en las sesiones plenarias.

Listado de planillas de pago de dietas por sector.

Listado de retención de renta individual y por sector.

Otros:

i) Almacenamiento de Actas



Figura 87. Almacenamiento de actas y acuerdos AGU
Fuente: Elaboración propia

j) Formato de solicitud de uso de licencia

Ciudad Universitaria, __ de ____ del 2018.

Ing. José María Sánchez Cornejo
Director de la Escuela de Ingeniería de Sistemas Informáticos
Facultad de Ingeniería y Arquitectura
Universidad de El Salvador.
Presente.

Respetable Ingeniero:

Por este medio, y en mi calidad de **Puesto del Remitente** y representante de **Nombre de la Contraparte** , que habiendo concluido el Trabajo de Graduación denominado: **SISTEMA INFORMÁTICO PARA EL APOYO DE REUNIONES Y CONTROL DE ACUERDOS DE LA ASAMBLEA GENERAL UNIVERSITARIA DE LA UNIVERSIDAD DE EL SALVADOR, SIARCA_AGU_UES**, desarrollado por los estudiantes:

Wendy Carolina Criollo Hernández,

Jaime Fermín Donado Iraheta,

Jonatan Benjamín López Henríquez,

Carlos Alberto Noyola Sánchez;

Y de conformidad con los procedimientos de la Facultad de Ingeniería y Arquitectura, **solicitamos la Licencia de Uso** correspondiente, así como la documentación relacionada.

Esperando una respuesta favorable a nuestra solicitud, nos despedimos.

Atentamente,

f. _____
Nombre del Representante
Nombre del Puesto
Nombre de la Contraparte

Nota: Incluir teléfono y correo electrónico de contacto del remitente.

k) Mapa de navegación del sitio

