

**UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA**



TRABAJO DE GRADO

**SISTEMA PARA LA GESTIÓN DE PROCESOS DEL ÁREA DE SOPORTE Y
TECNOLOGÍA INFORMÁTICA DE LA FACULTAD MULTIDISCIPLINARIA DE
OCCIDENTE DE LA UNIVERSIDAD DE EL SALVADOR**

**PARA OPTAR AL GRADO DE
INGENIERO(A) DE SISTEMAS INFORMÁTICOS**

PRESENTADO POR

**EDUARDO ACEVEDO FLORES
CARLOS EDUARDO FUENTES GUERRERO
LIDIA VERALICE LINARES OLMEDO
ANA GABRIELA QUEZADA TORRES**

DOCENTE ASESOR

INGENIERO LUIS ALONSO BARRERA FLORES

OCTUBRE, 2020

SANTA ANA, EL SALVADOR, CENTROAMÉRICA

UNIVERSIDAD DE EL SALVADOR
AUTORIDADES



M.Sc. ROGER ARMANDO ARIAS ALVARADO
RECTOR

DR. RAÚL ERNESTO AZCÚNAGA LÓPEZ
VICERRECTOR ACADÉMICO

ING. JUAN ROSA QUINTANILLA QUINTANILLA
VICERRECTOR ADMINISTRATIVO

ING. FRANCISCO ANTONIO ALARCÓN SANDOVAL
SECRETARIO GENERAL

LICDO. LUIS ANTONIO MEJÍA LIPE
DEFENSOR DE LOS DERECHOS UNIVERSITARIOS

LICDO. RAFAEL HUMBERTO PEÑA MARIN
FISCAL GENERAL

FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
AUTORIDADES



M.Ed. ROBERTO CARLOS SIGÜENZA CAMPOS
DECANO

M.Ed. RINA CLARIBEL BOLAÑOS DE ZOMETA
VICEDECANA

LICDO. JAIME ERNESTO SERMEÑO DE LA PEÑA
SECRETARIO

INGENIERO DOUGLAS GARCÍA RODEZNO
JEFE DEL DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA

Agradecimientos

Como equipo de trabajo de graduación agradecemos el apoyo y colaboración a las siguientes personas:

Primeramente, a Dios Todopoderoso y La Virgen María por habernos permitido culminar con éxitos los estudios universitarios, por ayudarnos a superar todas las pruebas que se presentaron a lo largo de la carrera.

Nuestros padres, por siempre haber brindado su apoyo y comprensión incondicional, por ser una guía y fortaleza para seguir adelante en cada decisión que hemos tomado. Agradecer a cada miembro de nuestra familia que de una forma contribuyó al éxito de este trabajo y finalización de la carrera.

Amigos y compañeros de la carrera, con quienes superamos los obstáculos académicos. A todos los docentes que contribuyeron en la formación académica. A nuestro asesor de tesis el Ing. Luis Barrera que con sus correcciones y paciencia nos ha guiado a través de todo este trabajo.

Compañeros de tesis, agradecer por el esfuerzo, entrega y dedicación para llegar hasta el final de este proceso.

INDICE

Introducción	xi
Capítulo I: Planteamiento del problema	12
1.1. Antecedentes.....	13
1.2. Formulación del Problema.....	14
1.2.1. Análisis Causa-Efecto.....	14
1.2.2. Análisis del Problema.....	15
1.2.3. Definición del Problema.....	17
1.3. Justificación.....	18
1.3.1. Análisis Estadístico.....	18
1.3.2. Importancia.....	23
1.3.3. Evaluación del Impacto Social.....	24
1.4. Objetivos.....	25
1.5. Alcances.....	26
1.6. Limitantes.....	27
Capítulo II: Marco Teórico	29
2.1. Antecedentes.....	30
2.1.1. Historia de la Universidad de El Salvador.....	30
2.1.2. Historia del Departamento de Ingeniería y Arquitectura.....	31
2.1.3. Ubicación Geográfica del Departamento de Ingeniería.....	33
2.2. Conceptos y generalidades del Soporte Técnico.....	33
2.2.1 Marco conceptual.....	33
2.2.2. Generalidades del Soporte Técnico.....	36
2.2.3. Importancia del Soporte Técnico.....	37
2.3. Modelo del Área de Soporte.....	37
2.3.1. Servicio de Soporte.....	38
2.3.2. Descripción General del Proceso de Soporte Técnico.....	39
2.3.3. Revisión de la Administración y Vigilancia.....	42
2.3.4. Organización de un Centro de Servicio.....	43
2.3.5. Modelo de Respuesta Gestionada.....	48

2.3.6. Descripción de la situación actual en el Área de Soporte Técnico y Tecnología Informática.	49
2.3.7. Como un Centro de Mantenimiento Informático.	51
2.3.8. Administración de registros.....	53
Capítulo III: Diseño Metodológico.....	54
3.1. Introducción.....	55
3.2. Enfoque de la Investigación.	55
3.3. Tipo de Investigación.	55
3.4. Diseño de Investigación.	56
3.5. Recursos y Herramientas de la Investigación.	58
3.5.1. Técnicas de Diagnóstico y Formulación del Problema.	58
3.5.2. Técnicas de Análisis y Diseño.	60
3.5.3. Metodología de Diseño de Bases de Datos.....	64
3.5.4. Descripción de Herramientas.....	65
3.6. Técnicas de Recolección de Datos.....	66
3.7. Cronograma de Actividades.....	68
3.8. Presupuesto del Proyecto.	69
Capítulo IV: Desarrollo	71
4.1. Especificación de requerimientos de software.	72
4.1.1. Introducción.	72
4.1.2. Descripción general.	73
4.1.3. Requisitos específicos.	78
4.1.3.4. Otros requisitos.....	82
4.2. Definición Tecnológica.	84
4.2.1. Comparativa entre tecnologías.	84
4.2.2. Tecnologías complementarias.	89
4.3 Diseño del software.....	90
4.3.1. Técnicas de programación.....	90
4.3.2. Diseño Arquitectónico.	92
4.4. Diseño de la base de datos.....	95
4.4.1. Diagrama Entidad – Relación.	95
4.4.2. Diccionario de Datos.....	96
4.5. Diagrama de clases.	120

4.6. Diagramas de Caso de Uso	122
4.7. Metodología de desarrollo.....	129
4.7.1. Modelo de desarrollo evolutivo.	131
4.7.2. Método de prototipos.	131
4.7.3. Etapas de desarrollo de software para métodos de prototipo.	132
4.8 Estándares de Programación.....	133
4.8.1 Nombre de Variables.	133
4.8.2 Variables Constantes.....	134
4.8.3 Métodos.....	134
4.8.4 Clases.....	134
4.8.5 Paquetes.....	134
Capítulo V: Conclusiones y Recomendaciones.....	136
5.1 Conclusiones.....	137
5.2 Recomendaciones.	139
Referencias Bibliográfica.....	140
ANEXOS	143
Anexo A	144

Índice de tablas

Capítulo I: Planteamiento del problema

TABLA 1.1: Mantenimientos Correctivos por Mes para el año 2017.....	18
TABLA 1.2: Mantenimientos Correctivos por Unidad/Departamento para el año 2017..	19
TABLA 1.3: Mantenimientos Preventivos por Unidad/Departamento para el año 2017.	21

Capítulo III: Diseño Metodológico

TABLA 3.1: Elementos de Diagrama de Caso de Uso.....	62
TABLA 3.2: Elementos de Diagrama de Clase.....	63
TABLA 3.3: Herramientas.....	65
TABLA 3.4: Presupuesto de salarios del personal.....	69
TABLA 3.5: Presupuesto de materiales gastables en la investigación.....	69
TABLA 3.6: Características de los equipos utilizados en la investigación	69
TABLA 3.7: Presupuesto de gastos directos en la investigación.....	69
TABLA 3.8: Presupuesto total de la investigación.....	70

Capítulo IV: Desarrollo

TABLA 4.1: Perfil del Administrador general del Sistema.....	76
TABLA 4.2: Perfil del Técnico de Área.....	76
TABLA 4.3: Perfil del Solicitante.....	77
TABLA 4.4: Interfaces de usuario	78
TABLA 4.5: Interfaces de Software.....	79
TABLA 4.6: Interfaces de comunicación	79
TABLA 4.7: Interfaces de adaptación al sitio	80
TABLA 4.8: Requerimientos funcionales generales	81
TABLA 4.9: Requerimientos de calidad	82
TABLA 4.10: Requerimientos de Hardware	83
TABLA 4.11: Requerimientos Mínimos de Máquina Cliente	83
TABLA 4.12: Comparativa entre lenguajes de programación.....	85

TABLA 4.13: Comparativa entre gestores de base de datos	88
TABLA 4.14: Modelo caso de uso - Ver estado	123
TABLA 4.15: Modelo caso de uso - Notificar solicitud.....	123
TABLA 4.16: Modelo caso de uso - Emitir solicitud	124
TABLA 4.17: Modelo caso de uso - Registrar servicio	124
TABLA 4.18: Modelo caso de uso - Generar elemento	125
TABLA 4.19: Modelo caso de uso - Agregar elemento.....	126
TABLA 4.20: Modelo caso de uso - Ver estado	127
TABLA 4.21: Modelo caso de uso - Buscar elemento	128
TABLA 4.22: Comparativa entres metodologías de desarrollo.....	130

Índice de gráficas

Capítulo I: Planteamiento del problema

GRÁFICA 1.1: Mantenimientos Correctivos por Mes para el año 2017.....	19
GRÁFICA 1.2: Mantenimientos Correctivos por Unidad/Departamento para el año 2017.....	20
GRÁFICA 1.3: Mantenimientos Preventivos Por Unidad/Departamento para el año 2017.....	22

Índice de figuras

Capítulo I: Planteamiento del problema

FIGURA 1.1: Diagrama Causa-Efecto.....	15
FIGURA 1.2: Diagrama de Caja Negra.....	15

Capítulo II: Marco Teórico

FIGURA 2.1: Diagrama General Del Soporte Técnico.....	40
FIGURA 2.2: Modelo De Respuesta Inmediata.....	45
FIGURA 2.3: Modelo Respuesta Inmediata: Múltiples Grupos De Recursos.....	45
FIGURA 2.4: Modelo Multitier Con Múltiples Grupos.....	47
FIGURA 2.5: Modelo de respuesta gestionada.....	48

Capítulo III: Diseño Metodológico

FIGURA 3.1: Diagrama de Ishikawa.....	59
FIGURA 3.2: Método de la Caja Negra	60
FIGURA 3.3: Diagrama de Caso de Uso	62
FIGURA 3.4: Diagrama de Clase.....	63
FIGURA 3.5: Bosquejo Diagrama Entidad Relación	65

Capítulo IV: Desarrollo

FIGURA 4.1: Modelo de Tres Capas	93
FIGURA 4.2: Modelo Vista Controlador.....	94
FIGURA 4.3: Diagrama ER.....	95
FIGURA 4.4: Diagrama de Clases de Tabla Equipos	121
FIGURA 4.5: Diagrama general de caso de uso.....	122
FIGURA 4.6: Diagrama de caso de uso - Generar elemento.....	125
FIGURA 4.7: Diagrama de caso de uso - Agregar elemento	126
FIGURA 4.8: Diagrama de caso de uso - Asignar elemento.....	127
FIGURA 4.9: Diagrama de caso de uso - Buscar elemento	128
FIGURA 4.10: Ciclo de desarrollo - Método de prototipos	133

Introducción

La Facultad Multidisciplinaria de Occidente, como cualquier ente, crece en su población estudiantil y administrativa; debido a la demanda con la que cuenta ha experimentado un aumento en el uso e incorporación de numerosos equipos informáticos, con los cuales los centros de cómputo han sido renovados, facilitando el día a día del personal administrativo, docentes y estudiantes que hacen uso de ellos. Esto ha generado la necesidad de delegar la tarea del mantenimiento informático a un grupo de personas que tienen conocimientos en el área, quienes son los encargados de brindar soporte y dar reparación a los equipos, con el fin de aumentar el tiempo de vida útil y garantizar un buen funcionamiento.

Sin embargo, gestionar todos los procesos de mantenimiento que se realizan a los equipos informáticos, genera la necesidad de llevar un control detallado de los servicios de soporte técnico que se han brindado. Por este motivo, se plantea la iniciativa de desarrollar un sistema informático que cumpla con las necesidades de llevar un detalle de los mantenimientos y el equipo que se ha revisado registrando sus especificaciones de hardware, sistema operativo, licencias de software y números de inventario. Así como ser capaz de generar reportes de estos, logrando la reducción en los tiempos y haciendo más sencilla la labor del personal, agilizando su trabajo.

El sistema a su vez mejorará la comunicación entre los interesados en solicitar un servicio y el Área de Soporte y Tecnología Informática, por medio de una plataforma que permita la interacción indirecta entre ambas partes, también le permitirá al solicitante monitorear en qué fase del proceso se encuentra su solicitud.

Además, facilitará al personal el almacenamiento de la calendarización de los mantenimientos preventivos. Generará bitácoras, estadísticas y reportes haciendo uso de los datos registrados en el sistema, todo esto hace que el trabajo se haga de una manera más rápida a los procesos manuales que se suelen llevar.

Capítulo I: Planteamiento del problema

1.1. Antecedentes.

En la Facultad Multidisciplinaria de Occidente debido a la creciente población estudiantil, el desarrollo de las metodologías de enseñanza y procesos administrativos, ligados al avance de la tecnología, siendo esta parte fundamental en los métodos cotidianos de aprendizaje y trabajo del día a día, la institución se vio en la necesidad de invertir en la compra de equipo informático para satisfacer esta creciente necesidad y así ofrecer un mejor servicio a la comunidad estudiantil.

En sus inicios no se contaba con un espacio físico apropiado ni con las herramientas necesarias para solventar problemas relacionados al mantenimiento y soporte del equipo adquirido, el cual con el uso y el paso del tiempo tiende a presentar problemas y deteriorarse si no se le da la atención necesaria. Es entonces que surge la necesidad de contar con personal capacitado, el cual sea encargado de apoyar la configuración, soporte y mantenimiento de los equipos informáticos, de esta forma estos funcionen de manera óptima, prolongando así su vida útil.

En un principio el Área solo contaba con el apoyo de un técnico quien poseía el conocimiento necesario para ayudar solventar cualquier problema que se presentará, él era el encargado de velar por las necesidades del cuerpo estudiantil, docente y administrativo, sin embargo, con el paso del tiempo se fue adquiriendo mayor cantidad de equipo, lo cual implicaba más esfuerzo y atención que la que se le podía dar en ese momento. El crecimiento fue a tal grado que la Facultad se vio en la necesidad de crear un Área especializada en soporte y mantenimiento de este. El Área inició formalmente en el año 2004, para ese entonces ya se contaba con un equipo de trabajo el cual se conformaba por estudiantes, los cuales ayudaban al técnico a cargo, distribuyendo la carga de trabajo en unidades o zonas por estudiante, estos se ayudaban con herramientas apropiadas para poder brindar el servicio de manera adecuada y de calidad.

Conforme el paso del tiempo se establecieron procesos y procedimientos para atender las solicitudes de las diferentes Áreas y Departamentos con los cuales cuenta la Facultad, a su vez se fueron incorporando métodos que facilitaban el trabajo del personal del Área.

Entre las actividades que se encarga de realizar el Área se encuentran las enfocadas al mantenimiento y reparación del equipo informático además de brindar apoyo en la instalación de nueva infraestructura de red y de igual manera brindar el mantenimiento de esta.

Hoy en día el Área sigue teniendo como finalidad asegurar el funcionamiento eficiente y continuo de las instalaciones y equipos, mediante la prevención, corrección y actualización de estos, con el fin de prolongar la vida útil, maximizar la seguridad de operación y optimizar los costos, todo esto a través de la programación del mantenimiento preventivo y correctivo.

1.2. Formulación del Problema.

1.2.1. Análisis Causa-Efecto.

Las causas identificadas que inciden en el problema se clasifican en dos categorías que se presentan a continuación:

Procedimientos

- Actualmente las solicitudes de servicio de soporte se realizan por medio de llamadas telefónicas y en persona, en las cuales se detalla lo que se necesita.
- Los registros de las solicitudes son llevados de forma manual, llenando un formulario impreso donde describe la falla. En esta misma hoja se encuentra el detalle del mantenimiento realizado.
- La información de los registros en papel es almacenada en hojas de cálculo.

Recursos Humanos

- El Área cuenta con poco personal para la realización de las tareas de soporte técnico y mantenimiento.

Estas causas o factores generan que el Área se retrase en sus actividades programadas, se acumulen las solicitudes de soporte técnico y los procesos manuales de registro de los servicios sean tediosos. En la *FIGURA 1.1* se muestra el Diagrama Causa-Efecto.

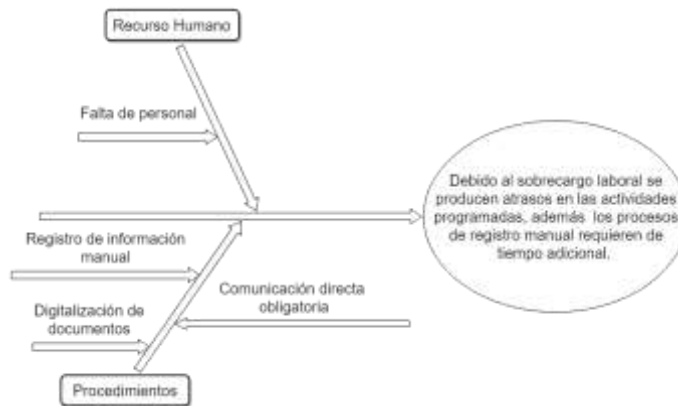


FIGURA 1.1: Diagrama Causa-Efecto

1.2.2. Análisis del Problema.

1.2.2.1. Método de la caja negra.

Para plantear el problema se utiliza el Método de la Caja Negra (Entrada – Proceso - Salida), en el cual el estado actual o entrada establece la problemática y el estado deseado establece una situación sin problema.



Registros y reportes manuales, consumo de tiempo, información almacenada física y digitalmente, comunicación directa requerida.

Registros y reportes automatizados, información consistente almacenada en una base de datos, comunicación directa opcional.

FIGURA 1.2: Diagrama de Caja Negra
(Jones, Christopher, Métodos de Diseño, Pág. 42)

Estado Actual

En el Estado Actual los problemas que se detectan son:

- El tiempo invertido para el manejo de la información es grande, debido al volumen de trabajo, en promedio 480 servicios correctivos y preventivos por año en el Área y al poco personal que se dedica a realizar este tipo de funciones.
- Debido al proceso manual de registro que se realiza, implica varias horas de trabajo para poder entregar un reporte cuando este sea necesario.
- Los datos registrados de los servicios brindados son almacenados en hojas de cálculo, lo que dificulta llevar un control preciso de la información.
- Es difícil conocer el historial de servicios que se han realizado a un determinado equipo en un período de tiempo.
- Debido a los 250 equipos en promedio con los que cuenta la Facultad es difícil organizar la programación de mantenimientos preventivos para todos.
- Cuando es necesario un mantenimiento correctivo, el solicitante debe esperar que el encargado del Área se encuentre disponible para atender la petición, debido a que la comunicación sólo puede ser cara a cara o vía telefónica.

Estado Deseado

Mediante el desarrollo de un Sistema Informático llegaremos al Estado Deseado, mejorando el control y agilizando el registro de los servicios que el Área brinda, para lo cual es necesario:

- Almacenar, clasificar y mantener actualizada la información de los equipos y los servicios que éstos han recibido.
- Búsqueda de información sobre los equipos, servicios que éstos han recibido y en qué Unidad o Departamento se realizó, según los criterios que el Área de Soporte y Tecnología considere necesarios.
- Proporcionar consistencia en los datos que permita tener congruencia en la información para todos los equipos de todas las Unidades y Departamentos.
- Mejorar el control de todos los mantenimientos preventivos que deben ser calendarizados para las Unidades y Departamentos, lo que permitirá que los

equipos puedan recibir oportunamente las reparaciones que sean necesarias para su buen funcionamiento.

- Generación de reportes y estadísticas según criterios establecidos por el encargado del Área de Soporte Técnico.

1.2.2.2. Variables de Solución.

Las variables de solución son todos aquellos elementos que ayudarán en la transformación del estado actual al estado deseado, entre estos se tienen:

- Recurso humano necesario
- Requerimientos de Desarrollo
- Requerimientos de Operación
- Requerimientos de Implementación

1.2.3. Definición del Problema.

De acuerdo con la descripción del estado actual y el estado deseado, se identifica que el problema principal es el consumo de tiempo necesario en los procesos que requieran registrar datos y gestionar información para la generación de reportes, pues estos se realizan de manera manual.

1.2.4. Propuesta de solución.

En base a las herramientas utilizadas para el análisis podemos afirmar que: Ante la necesidad de contar con un proceso de registro de datos automatizados en el Área, es necesario un sistema informático que apoye las actividades administrativas y de control, permitiendo obtener información oportuna que ayude en la generación de reportes y estadísticas que faciliten la toma de decisiones; también mejorando la comunicación entre las partes (Solicitantes-Área, Área-Junta Directiva).

1.3. Justificación.

Actualmente el Área de Soporte y Tecnología de la Facultad Multidisciplinaria de Occidente no cuenta con un Sistema Informático que le permita registrar, actualizar y consultar la información relacionada con los mantenimientos correctivos y preventivos que sea realizan en los equipos de la Facultad, por lo cual es necesario contar con uno que permita capturar, procesar y obtener reportes de los servicios que se registren a lo largo del tiempo.

1.3.1. Análisis Estadístico.

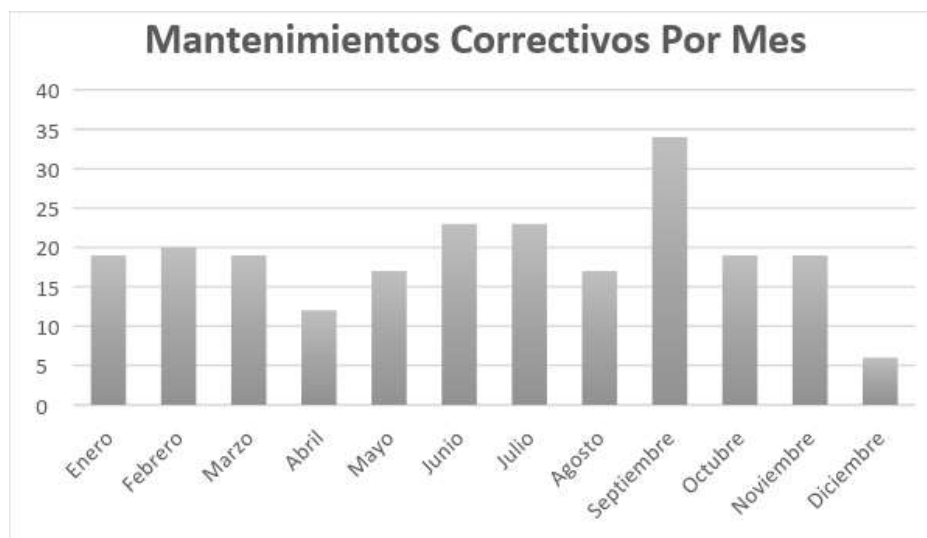
La Facultad cuenta con aproximadamente 250 equipos informáticos, los cuales ayudan a los Departamentos y Unidades a llevar a cabo sus labores, ya sea en áreas administrativas o a los estudiantes. Cada uno de estos equipos cuenta con todas sus respectivas especificaciones, mediante las cuales se lleva un registro de los fallos que han tenido durante su vida útil.

Para el año 2017, se tiene registro que se realizaron los siguientes mantenimientos correctivos en toda la Facultad:

Mes	Cantidad
Enero	19
Febrero	20
Marzo	19
Abril	12
Mayo	17
Junio	23
Julio	23
Agosto	17
Septiembre	34
Octubre	19
Noviembre	19
Diciembre	6
Total	228

TABLA 1.1: Mantenimientos Correctivos por Mes para el año 2017
(Estadísticas proporcionadas por el Área de Soporte y Tecnología)

La *GRÁFICA 1.1* muestra todos los mantenimientos correctivos que se realizaron por mes en toda la Facultad:

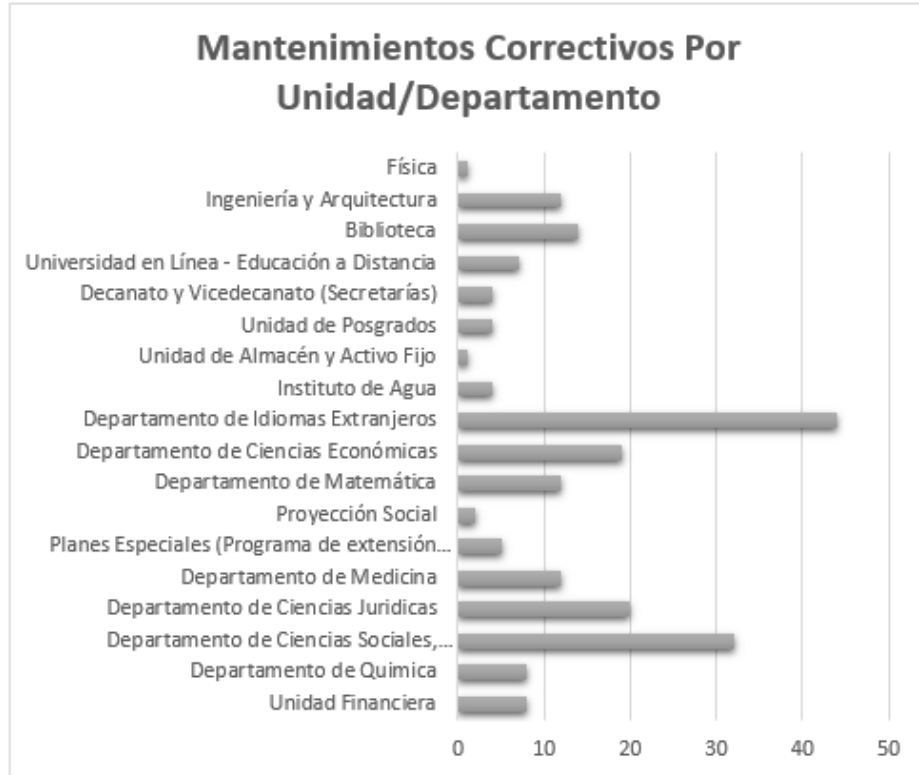


GRÁFICA 1.1: Mantenimientos Correctivos por Mes para el año 2017
(Estadísticas proporcionadas por el Área de Soporte y Tecnología)

La TABLA 1.2 y GRÁFICA 1.2 muestran el detalle de los mantenimientos correctivos realizados en cada Departamento:

Unidad	Cantidad
Unidad Financiera	8
Departamento de Química	8
Departamento de Ciencias Sociales, Filosofía y Letras	32
Departamento de Ciencias Jurídicas	20
Departamento de Medicina	12
Planes Especiales (Programa de extensión de Idiomas)	5
Proyección Social	2
Departamento de Matemática	12
Departamento de Ciencias Económicas	19
Departamento de Idiomas Extranjeros	44
Instituto de Agua	4
Unidad de Almacén y Activo Fijo	1
Unidad de Posgrados	4
Decanato y Vicedecanato (Secretarías)	4
Universidad en Línea - Educación a Distancia	7
Biblioteca	14
Ingeniería y Arquitectura	12
Física	1
Total	228

TABLA 1.2: Mantenimientos Correctivos por Unidad/Departamento para el año 2017
(Estadísticas proporcionadas por el Área de Soporte y Tecnología)



GRÁFICA 1.2: Mantenimientos Correctivos por Unidad/Departamento para el año 2017 (Estadísticas proporcionadas por el Área de Soporte y Tecnología)

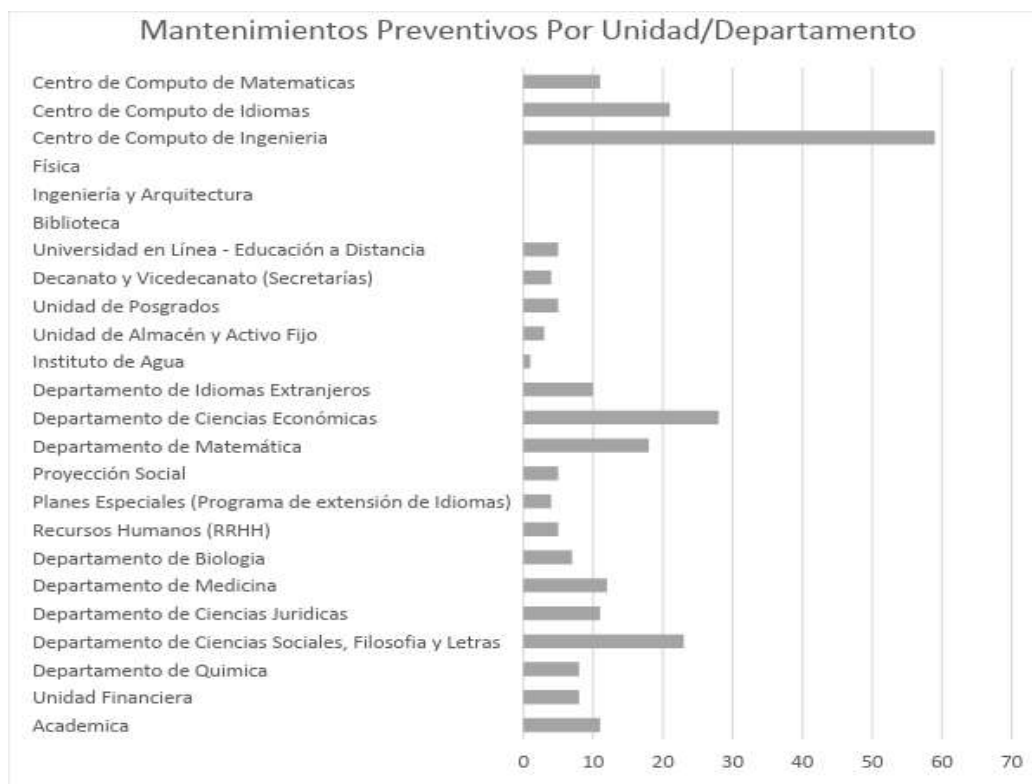
Los mantenimientos correctivos se realizan para corregir defectos o fallas observadas en los equipos informáticos. Dichas fallas se pueden presentar en cualquier momento y sin previo aviso, lo cual provoca que un equipo pueda quedar inutilizable hasta su reparación. Con el proceso actual de control, cuando se genera un incidente es necesario que el solicitante contacte directamente al personal del Área, ya sea por conversación cara a cara o vía telefónica, es decir, la pronta notificación del problema depende de la disponibilidad del personal para recibirla, eso sin contar el tiempo necesario para la reparación. También es necesario tomar en cuenta que la cantidad de mantenimientos correctivos tiende a aumentar debido a la antigüedad de los equipos. Para este caso, el Área de Soporte y Tecnología Informática durante el año 2017 tuvo que realizar 228 mantenimientos correctivos en todo el año, un promedio de 20 servicios por mes. Esta cantidad de servicios tuvieron que ser registrados de forma manual en hojas de papel y luego ser trasladados a hojas de cálculo, lo que podría darse el caso de contribuir a la generación de errores

al momento de elaborar los reportes correspondientes y además existe la necesidad de una ubicación física para los documentos, lo que puede llevar a la pérdida y degradación de la papelería con el paso del tiempo.

Aparte de los mantenimientos correctivos, el Área se encarga de realizar mantenimientos preventivos, para el año 2017 se tiene registro que se realizaron los siguientes en la Facultad (*TABLA 1.3* y *GRÁFICA 1.3*):

Unidad	Cantidad
Académica	11
Unidad Financiera	8
Departamento de Química	8
Departamento de Ciencias Sociales, Filosofía y Letras	23
Departamento de Ciencias Jurídicas	11
Departamento de Medicina	12
Departamento de Biología	7
Recursos Humanos (RRHH)	5
Planes Especiales (Programa de extensión de Idiomas)	4
Proyección Social	5
Departamento de Matemática	18
Departamento de Ciencias Económicas	28
Departamento de Idiomas Extranjeros	10
Instituto de Agua	1
Unidad de Almacén y Activo Fijo	3
Unidad de Posgrados	5
Decanato y Vicedecanato (Secretarías)	4
Universidad en Línea - Educación a Distancia	5
Biblioteca	0
Ingeniería y Arquitectura	0
Física	0
Centro de Cómputo de Ingeniería	59
Centro de Cómputo de Idiomas	21
Centro de Cómputo de Matemáticas	11
Total	259

TABLA 1.3: Mantenimientos Preventivos por Unidad/Departamento para el año 2017
(Estadísticas proporcionadas por el Área de Soporte y Tecnología)



GRÁFICA 1.3: Mantenimientos Preventivos Por Unidad/Departamento para el año 2017 (Estadísticas proporcionadas por el Área de Soporte y Tecnología)

Los mantenimientos preventivos se realizan con la finalidad de garantizar el buen funcionamiento, la fiabilidad y alargar la vida útil de los equipos informáticos. Como se puede observar, para el año 2017 se realizaron 259 mantenimientos de este tipo, un promedio de 22 por mes, los cuales, al igual que los mantenimientos correctivos, tienen que ser registrados de manera manual en hojas de papel y luego ser trasladados a hojas de cálculo para permitir llevar un mínimo control, más todos los inconvenientes antes mencionados.

Hay que tener en cuenta que, para notificar la realización de un mantenimiento preventivo, el Área envía un documento para establecer la fecha en que éste se llevará a cabo, lo que se suma al problema de la comunicación entre el encargado y las personas que reciben el servicio, pues depende de la disponibilidad de ambas partes para enviar y recibir dichos documentos, aparte que aumenta la cantidad de papel que debe ser almacenada en el Área.

El documento de “correspondencia” que el Área envía tiene un mínimo de 3 páginas de papel, durante el año 2017 se tuvo un total de 112 páginas utilizadas para este fin. Si consideramos el papel utilizado para los mantenimientos preventivos y correctivos, se tiene que:

- Mantenimientos Correctivos: 228
- Mantenimientos Preventivos: 168 + 3 (1 por cada Centro de Cómputo)
- Correspondencia: 112

Se tuvo un gasto aproximado de 508 páginas durante el año 2017, las cuales deben ser almacenadas físicamente y verificadas manualmente para fines estadísticos y generación de reportes.

A raíz de esta situación es muy difícil para el personal del Área realizar búsquedas eficientes de información específica que consideren necesarias y/o importantes, tales como el historial de mantenimientos de un único equipo, el historial de todos los mantenimientos en un rango de fechas o el historial de mantenimientos por Unidad o Departamento.

El sistema informático propuesto permitirá la creación de una base de datos con la información de los mantenimientos y datos sobre los equipos, contribuyendo a un mejor y eficiente control de la información.

1.3.2. Importancia.

Al desarrollar un Sistema, contribuirá a solventar la situación planteada con anterioridad, ya que optimizará la forma en la cual se lleva a cabo la gestión de las bitácoras, así como un registro de equipos con fallas técnicas, además ayudaría a la comunicación entre los solicitantes y el personal del Área, quienes reciben peticiones a diario para revisar incidencias en equipos de Departamentos y Unidades Administrativas, lo que evitaría retrasos en la prestación del servicio.

Además, con el sistema informático se facilitará la entrega de información solicitada por los diferentes Departamentos y Unidades, ya que la automatización en el procesamiento de datos minimizará el tiempo empleado.

El sistema disminuirá irregularidades y errores en la información registrada, de esta forma será almacenada en el momento que se solicite la tarea, ayudando a llevar un mejor control de los mantenimientos, la calendarización del mantenimiento preventivo también se verá mejorada.

También contará con la realización de reportes y permitirá que el encargado pueda crear respaldos de la información, además se presentará de forma más estructurada y completa, logrando así que la realización de estadísticas sea más eficiente.

1.3.3. Evaluación del Impacto Social.

La evaluación del impacto social pretende identificar el grado de incidencia que tendrá la implementación del Sistema en la situación actual de la Facultad Multidisciplinaria de Occidente de la Universidad de El Salvador.

A continuación, se presentan las entidades en las cuales el Sistema Informático tendrá algún grado de incidencia cuando sea implementado.

1.3.3.1. Personal Solicitante.

Véase como solicitante a todo el personal individual administrativo o docente que tiene asignado un equipo informático. El Sistema, mediante la plataforma de solicitud mejorará la comunicación a la hora de requerir un servicio de soporte, dando una alternativa a las llamadas telefónicas o ir en persona. También podrán visualizar y conocer el estado del proceso.

1.3.3.2. Área de Soporte y Tecnología Informática.

Al implementar el Sistema en el Área, el tiempo que se tomaban en los procesos de registro manual de solicitud soporte técnico y mantenimiento se verá reducido significativamente, ya que se hace directamente en la interfaz y se almacena en la base de datos, gracias a esto se podrán enfocar en terminar otras tareas.

1.3.3.3. Facultad Multidisciplinaria de Occidente.

El Sistema para la gestión del soporte técnico en equipos, es un proyecto propuesto para su implementación con el objeto de innovar los procesos informáticos, dando paso a una comunicación no presencial que permita dar soporte al sector administrativo y docente de la Facultad Multidisciplinaria de Occidente de la Universidad de El Salvador, y contribuir con ello a mejorar el trabajo del principal centro de estudios de educación superior de toda la república salvadoreña.

1.4. Objetivos.

Objetivo General.

Desarrollar un Sistema para la gestión de los servicios brindados a las distintas Unidades y Departamentos, con el fin de automatizar los procesos de registro de información realizados en el Área de Soporte y Tecnología Informática de la Facultad Multidisciplinaria de Occidente de la Universidad de El Salvador.

Objetivos Específicos.

- Diseñar una base de datos para el registro de las solicitudes reportadas, almacenamiento de la calendarización de los mantenimientos preventivos y especificaciones técnicas de hardware y software de los equipos.
- Desarrollar una interfaz que le permitirá al solicitante reportar problemas y hacer un seguimiento del mantenimiento de su equipo, verificando el estado en que se encuentra.

- Generar la solicitud de mantenimiento de equipo informático de forma automática en base al reporte del problema ingresado por el usuario.
- Facilitar la generación de bitácoras, reportes y estadísticas por parte del administrador del sistema.

1.5. Alcances.

Para la ejecución de este Sistema Informático se toma en cuenta lo siguiente:

- La información que almacenará la base de datos:
 - Detalles del mantenimiento solicitado, los cuales son: Información del solicitante, descripción del equipo, detalle de la falla, fecha y hora de la solicitud, prioridad, información del técnico asignado.
 - Detalles del servicio realizado, los cuales son: Mantenimiento aplicado, detalle de reparación, fecha y hora de entrega.
 - Detalles de hardware y software del equipo informático: Especificaciones de memoria de RAM, capacidad de disco duro, componentes por equipo, sistema operativo, licencias, utilitarios y los que el Área considere importantes. Los equipos registrados serán identificados por el número de inventario asignado por la Facultad o por el TAG asignado por el fabricante.
 - Los datos para la calendarización de los mantenimientos preventivos deberán ser ingresados por el usuario.
- La interfaz para reportar problemas funcionará a través de un formulario, en el cual el usuario deberá ingresar su nombre, Departamento o Unidad y descripción de la falla. Cuando se envíe la solicitud, se generará un código que permitirá verificar el estado en que se encuentra la petición (pendiente, abierto, finalizado con detalles, finalizado).
- El formulario de mantenimiento cargará los datos del formulario de solicitud enviado, que además asignará el técnico que estará a cargo, definirá la

prioridad, detalle del mantenimiento (servicio) brindado, en base a la información contenida en la base de datos.

- Con la información almacenada en la base de datos se generarán los reportes, bitácoras y/o estadísticas, tomando los criterios que el Área considere importantes.

1.6. Limitantes.

- Falta de acceso a las especificaciones de los servidores, ya que son datos confidenciales.
- No se tiene acceso al alojamiento del Sistema en los servidores de la Facultad, lo cual limita el desarrollo adecuado, la estimación de costes e implementación del Sistema.

Capítulo II: Marco Teórico

2.1. Antecedentes.

2.1.1. Historia de la Universidad de El Salvador.

La Universidad de El Salvador fue fundada el 16 de febrero de 1841. La ejecución del decreto de fundación corrió a cargo del jefe de sección encargado del Ministerio de Relaciones y Gobernación, quien dispuso su impresión, publicación y circulación. La UES inició sus actividades hasta 1843, impartiendo matemáticas puras, lógica, moral, metafísica y física general. En 1880 se subdividió en facultades, algunas de las cuales desaparecieron tiempo después, mientras que otras nuevas fueron creadas. Es el primero y más grande centro de estudios superiores de la República de El Salvador. También es la única universidad pública del país. Ciudad Universitaria, su campus central, está ubicado en la ciudad de San Salvador; también cuenta con sedes en las ciudades de Santa Ana, San Miguel y San Vicente.

Anualmente la UES hace pública una convocatoria nacional de ingreso universitario en los distintos medios de comunicación, en esta se especifican los pasos a seguir, las sedes, advertencias y recomendaciones del proceso de selección. Los aspirantes a nuevo ingreso deben someterse al examen general de admisión donde se evalúan las materias básicas (Matemáticas y Álgebra, Ciencias Sociales, Ciencias Naturales y Lenguaje y Literatura). En 2013, según los datos basados en el número de estudiantes matriculados en el Alma Máter, se contabilizaron más de 56,000 estudiantes. (Universidad de El Salvador [UES], Nuestra Universidad, 2016).

Visión

“Ser una universidad transformadora de la educación superior y desempeñar un papel protagónico relevante, en la transformación de la conciencia crítica y prepositiva de la sociedad salvadoreña, con liderazgo en la innovación educativa y excelencia académica, a través de la integración de las funciones básicas de la universidad: la docencia la investigación y la proyección social”.

Misión

“Institución en nuestro país eminentemente académica, rectora de la educación superior, formadora de profesionales con valores éticos firmes, garante del desarrollo, de la ciencia, el arte, la cultura y el deporte. Crítica de la realidad, con capacidad de proponer soluciones a los problemas nacionales a través de la investigación filosófica, científica, artística y tecnológica; de carácter universal”.

Facultad Multidisciplinaria de Occidente.

La Facultad Multidisciplinaria de Occidente, como parte del sistema de la Universidad de El Salvador; administra los mismos planes y programas de estudio que lleva cada una de las Facultades de la unidad central.

Se convierte en Facultad Multidisciplinaria de Occidente a partir del Acuerdo del Consejo Superior Universitario No. 39-91-95-IX de fecha 4 de Junio de 1992. (UES, Acuerdos CSU, 1992). Lo que permite:

- Autonomía Académica y Administrativa.
- Ejecutar eficientemente los planes y programas propuestos.
- Planificar y desarrollar iniciativas de acuerdo con las necesidades de la zona.
- Grados Académicos que la Institución otorga:
 - Profesor.
 - Licenciado.
 - Ingeniero.
 - Doctor en Medicina.
 - Maestría.

2.1.2. Historia del Departamento de Ingeniería y Arquitectura.

La Facultad Multidisciplinaria de Occidente de la Universidad de El Salvador, es la más antigua de las facultades descentralizadas de la UES. Tiene su sede en la ciudad de Santa Ana. Es el principal centro de estudios de la zona occidental de El Salvador.

En el año de 1963, por iniciativa de la Sociedad de Abogados de Occidente, se solicitó al Rector de la UES, Dr. Fabio Castillo Figueroa, la fundación de un centro regional de estudios superiores, en Santa Ana, la ciudad más poblada de la zona occidental de El Salvador, con el fin de atender a la numerosa población estudiantil de los departamentos de Santa Ana, Ahuachapán y Sonsonate.

El 16 de julio de 1965, el Consejo Superior Universitario de la UES, autoriza la creación del Centro Universitario de Occidente, con sede en Santa Ana. Las clases se iniciaron en mayo de 1966. Al principio, sólo se impartieron las áreas comunes, pero en 1971, se empezaron a impartir carreras completas.

El Centro Universitario de Occidente, también sufrió la represión de la que fue víctima, la generalidad de la UES, siendo clausurado y ocupado militarmente entre 1972-1973 y nuevamente en 1980. Tras la reapertura en 1983, hubo un período de decadencia por la coyuntura política que vivía El Salvador.

El 4 de junio de 1992, el Centro Universitario de Occidente se transforma en Facultad Multidisciplinaria. En 2001-2002, los edificios del campus de Santa Ana fueron ampliados por iniciativa de la Rectora, María Isabel Rodríguez.

Misión

“Construir un Departamento cuyas actividades académicas, científicas, y tecnológicas, tengan un real y significativo impacto en el desarrollo educativo y, consecuentemente, en la calidad de vida de la población de la región occidental y del país en general, promoviendo valores socialmente positivos como: La democracia, la solidaridad, la tolerancia, el espíritu científico y la investigación al servicio de la población”.

Visión

“La formación de profesionales con alta calidad académica, científica y humana; capaz de incidir en la solución de los problemas de la región occidental y de El Salvador en general”.

2.1.3. Ubicación Geográfica del Departamento de Ingeniería.

La FMO, tiene su sede en la ciudad de Santa Ana. Se encuentra ubicada al Final de la Avenida Fray Felipe Sur, al sur del estadio Oscar Alberto Quiteño. Y en lo que respecta al Departamento de Ingeniería se encuentra en el segundo nivel del edificio de usos múltiples al costado norte del departamento de idiomas.

La infraestructura del Departamento de se encuentra formada por cubículos en forma rectangular y en cada uno de estos se encuentran ubicados los docentes del departamento de Ingeniería, en cada cubículo hay entre uno o dos docentes, en los cuales pueden guardar materiales diversos que les sirven para preparar y para impartir sus clases, tales como folletos, trabajos de los alumnos, libros, computadores, además este el lugar idóneo donde los alumnos pueden encontrar a los docentes cuando necesitan hacer una consulta.

2.2. Conceptos y generalidades del Soporte Técnico.

2.2.1 Marco conceptual.

Las siguientes definiciones de mantenimiento fueron aprobadas por el Comité Panamericano de Ingeniería de Mantenimiento (COPIMAN), las cuales serán analizadas conforme se mencionen en el texto.

Soporte técnico:

Es el área encargada de prestar servicios para el mantenimiento del equipo informático cuando este presenta errores o fallos. Este servicio es brindado por técnicos o personal con conocimientos afines al área.

Mantenimiento Correctivo:

Servicios de reparación en ítems con falla. Esta estrategia consiste en localizar averías o defectos y corregirlos o repararlos. La reparación de la falla se presenta como emergencia.

Mantenimiento Correctivo Software:

Se centra en las actividades a desarrollar cuando un programa informático falla, luego se realiza un estudio de las principales soluciones a los problemas de software más frecuentes que impiden el funcionamiento del equipo.

Mantenimiento Correctivo Hardware:

Servicios de reparación en ítems con falla. Aborda la interpretación de los pitidos que genera el equipo cuando hay problemas físicos que impiden su arranque, así también las principales soluciones a los problemas físicos más frecuentes que impiden el funcionamiento del equipo.

Mantenimiento Preventivo:

Servicio de reparación destinado a la conservación de equipos o instalaciones mediante la realización de revisión y reparación que garanticen su buen funcionamiento y fiabilidad.

Mantenimiento Preventivo de Software:

Plantea las acciones periódicas que se deben realizar sobre las cuentas de usuario, los discos de almacenamiento, el sistema operativo, el antivirus y la realización de Back-ups; reseña algunas herramientas que facilitan el monitoreo y revisión del software en el equipo.

Mantenimiento Preventivo Hardware:

Describe las actividades de limpieza de partes, conexión de periféricos e inventario que se recomienda se realice periódicamente a los equipos.

Tecnología Informática:

Según lo definido por la asociación de la Tecnología Informática de América (ITAA), es el estudio, diseño, desarrollo, innovación puesta en práctica, ayuda o gerencia de los sistemas informáticos computarizados, particularmente usos del software.

Ítem:

Término general para indicar un equipo, obra o instalación.

Falla:

Finalización de la habilidad de un ítem para desempeñar una función requerida.

Defecto:

Evento en los equipos que no impide su funcionamiento, pero a corto o largo plazo puede provocar su indisponibilidad.

Error:

Eventos en los operadores que aún no impiden el buen funcionamiento del equipo, pero que pueden a corto o largo plazo provocar su indisponibilidad. (Esta definición es propuesta por el autor al no encontrar otra adecuada; pues esta sugiere que el defecto es al computador como el error es al operario. Ambos, defecto y error pueden llevar el equipo a la falla si no se atienden de manera oportuna).

Para esta investigación se usarán las palabras problema o falla para englobar los términos falla, defecto y error indistintamente.

2.2.2. Generalidades del Soporte Técnico.

2.2.2.1. Ventajas del Soporte Técnico.

- Reducción de fallas: El riesgo de alguna falla en equipos informáticos es disminuido considerablemente.
- Confiabilidad: Los equipos operan en mejores condiciones, ya que se conoce su estado, y funcionamiento. Mejora sustancialmente la fiabilidad de los equipos.
- Mayor duración de los equipos e instalaciones. Prolonga la vida útil de los equipos.
- Control y planeación: Facilita el poder llevar una mejor planeación y un mejor control sobre el mantenimiento que debe ser aplicado tanto en los dispositivos como en las instalaciones
- Uniformidad en la carga de trabajo para el personal del mantenimiento debido a una programación de actividades.
- Reduce la probabilidad de paros imprevistos. menor tiempo de inactividad no planificado causado por falla de los equipos. Se generan menos errores en las operaciones cotidianas.
- Permite mejorar el control sobre el funcionamiento del equipo y su productividad, así como la programación del mantenimiento que será aplicado en este.

2.2.2.2. Desventajas del Soporte Técnico.

Entre sus pocas desventajas se encuentran:

- Dado que las labores de mantenimiento de los equipos se efectúan con cierta periodicidad, no permiten que se pueda determinar exactamente la depreciación o desgaste de los equipos.
- Es necesario que el personal encargado del mantenimiento cuente con experiencia en el área.

2.2.3. Importancia del Soporte Técnico.

En la mayoría de pequeñas y medianas empresas no se dispone de un encargado o un área de soporte técnico que se encargue de realizar el mantenimiento de todo el “parque” informático de la misma. Se recurre a este servicio sólo cuando existe algún problema que es grave y necesita ser resuelto con urgencia.

Ocurre con demasiada frecuencia que se parte de un pequeño problema y se evita resolverlo a tiempo, lo que al final acaba convirtiéndose en un enorme y grave error. Solo en estos casos, los gerentes o administradores de las instituciones se dan cuenta de lo importante que es disponer de soporte técnico para solventar estos problemas.

Cualquier sistema informático, por sencillo que sea, necesita un mantenimiento mínimo. La finalidad de dichos mantenimientos es conseguir que los equipos sean operativos el mayor tiempo posible y que, durante ese tiempo, funcionen sin fallos. Es importante brindar un correcto soporte técnico por varias razones, entre las que se pueden destacar:

- A medio y largo plazo el coste es menor, ya que siempre será más económico mantener un equipo para evitar fallos que reemplazarlo cuando está averiado.
- Un fallo en el sistema puede provocar pérdidas de información de costos incalculables.

Se necesita seguridad informática por los datos importantes que se almacenan en los equipos y la pérdida de tiempo que sucede cuando los sistemas no funcionan.

2.3. Modelo del Área de Soporte.

En términos generales, la administración de problemas es una función de negocios conformada por personal, procesos y herramientas organizadas que se usan para resolver inconvenientes de clientes. Esta función tradicionalmente es

responsabilidad y administrada por un “servicio de asistencia”, “servicio de ayuda” o “servicio de soporte técnico”. Con las compañías incrementando drásticamente sus inversiones en tecnología, la visibilidad de un servicio de soporte técnico y la presión en el desempeño de este incrementan en la misma medida. La amplitud y profundidad de experiencia requerida ahora para dar soporte a una infraestructura común es prácticamente inimaginable comparada al de una mainframe de 1970s o de 1980s. En la actualidad también se ha incrementado su dificultad de dar soporte a clientes externos, debido a la complejidad de sus infraestructuras y los servicios que demandan constantemente. Muchos clientes esperan servicios “on-demand” con acceso a variedad de opciones de dichos servicios (telefonía, internet, emails y más).

2.3.1. Servicio de Soporte.

El servicio de soporte es el encargado de ayudar a los clientes a usar los productos o servicios. Dicho servicio de soporte tiene los recursos más actualizados, disponibles para resolver problemas que los clientes encuentran cuando usan dichos productos o servicios. Los empleados de esta área cuentan con buenas habilidades de comunicación, habilidades técnicas, habilidades para resolver problemas, para poder identificar, rastrear y resolver las situaciones que los clientes presenten.

En las últimas décadas muchos de estos servicios de soporte han evolucionado del modelo tradicional de solo resolver problemas a ofrecer servicios con valores adicionales, llamados “centros de servicio” (Walker, Gary. IT Problem Management, 2001). Un centro de servicio ofrece servicios del modelo tradicional para resolver problemas, así como también ofrecen otros de valor agregado, como pueden ser ayudar en la compra de equipo nuevo para los clientes, coordinar y supervisar para establecer nuevos empleados o asistir en alguna toma de decisiones.

2.3.1.1. Centros de Servicio Internos y Externos.

Un centro de servicio puede tener clientes internos, externos y en algunos casos de ambos. Los clientes internos son empleados que trabajan para la compañía

a la que pertenece el centro de servicio. Los clientes externos son todos aquellos que compran productos o servicios a la compañía, por lo que necesitan soporte para utilizarlos adecuadamente. Los centros de servicio internos ayudan a los empleados a usar la infraestructura interna que la compañía ha puesto a su disposición para que realicen su trabajo diario. Muchas compañías tienen un centro de servicio dedicado a clientes internos y otro dedicado a clientes externos, aunque ambos se dediquen a los mismos tipos de procesos. Quizás, la principal diferencia radica en que los centros externos son lucrativos para las compañías.

Desde una perspectiva operacional, los centros externos usualmente tienen que dar soporte a una más amplia gama de productos y servicios que un centro de servicio interno. Este es el caso cuando los centros internos deben dar soporte a la variedad de tecnologías que se usan en toda la infraestructura de la compañía. Microsoft es un ejemplo de compañía que posee un centro de servicio separado para los clientes internos y externos.

2.3.2. Descripción General del Proceso de Soporte Técnico.

La administración de problemas es un proceso complejo. Muchas organizaciones han proveído servicios desde un centro de servicio formal durante años. Con el paso del tiempo, estos centros han mejorado en todos los aspectos de los servicios que brindan. Estas mejoras han sido en procesos, en desempeño, en costos y eficiencia general. Los centros de servicio conducen la industria del soporte técnico de tal modo que las herramientas de soporte actuales son superiores que las disponibles hace un par de años atrás.

A alto nivel, la administración de problemas se puede dividir en cinco procesos principales, como se muestra en la *FIGURA 2.1*:

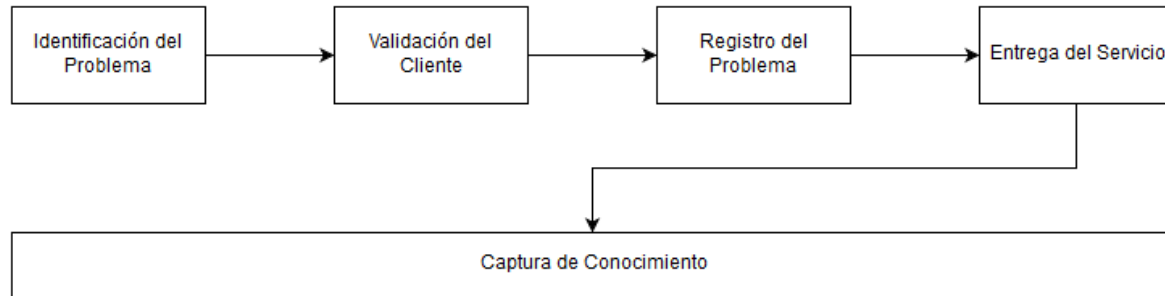


FIGURA 2.1: Diagrama General Del Soporte Técnico
(Walker, Gary. IT Problem Management, 2001)

2.3.2.1. Identificación del Problema.

Son todas las personas, procesos y herramientas usadas para detectar y reportar problemas. Los problemas pueden ser detectados antes o después de que ocurran (proactiva o reactivamente), se puede hacer mediante software de monitoreo, con ayuda de los empleados o los clientes. La meta es detectar tantos como sean posibles antes de que ocurran, para que puedan ser reparados antes que se conviertan en una situación que afecte la productividad (productivity-killing problem). También pueden ser detectados de antemano a través de análisis de tendencias y la observación. No importa cuántas herramientas y técnicas se utilicen, es imposible detectar todos los fallos con anticipación, es por esto por lo que los clientes siempre serán la primera fuente de detección.

2.3.2.2. Validación del Cliente.

Una vez que el problema ha sido reportado, el centro de servicio debe evaluar que provenga de un cliente válido. Posteriormente, se debe validar que la petición del cliente sea un servicio brindado por el centro, si no es el caso, debe ser referido donde sea necesario. Si tanto el cliente como la petición son válidos, el centro debe evaluar si es apto para recibir el servicio que está solicitando. Generalmente este proceso de validación se realiza en centros de servicio externos; esto tiene sentido pues generan cargos por soporte. Sin embargo, se está volviendo importante en los centros de servicio internos realizar este proceso de validación, esto es debido a que muchas empresas confían en su infraestructura técnica y tienen mucha información de la compañía en línea.

2.3.2.3. Registro del Problema.

Este proceso ocurre después que el problema ha sido detectado y validado. El registro aplica tanto para problemas que necesitan una solución como los servicios que brinda el centro (servicios de valor agregado). Una vez que una petición válida ha sido recibida, se debe crear un expediente de dicha petición. Este expediente, registro, o ticket es el documento básico usado para gestionar la petición desde el inicio hasta el fin. Este documento no solo sirve para determinar una transacción que necesita un trabajo sino también para generar estadísticas de desempeño. Dos importantes factores deben ser almacenados durante el proceso de registro: la prioridad de la petición y la categoría. Basados en la prioridad, el centro de servicio asigna recursos para completar el trabajo. La categoría asignada a un registro o ticket determina su forma de manejarse y hacia dónde se encamina (en caso de que sea necesario). Por ejemplo, si la petición es para reparar un problema de hardware, si es correctamente categorizada, dicha petición debería ser encaminada a los recursos que se encargan de manejar problemas de hardware. Más allá, las categorías son usadas para recoger estadísticas que permiten evaluar el tipo y el volumen de peticiones entrantes al centro de servicio.

2.3.2.4. Entrega del Servicio.

La entrega del servicio es el proceso de realizar el trabajo requerido para completar la petición y entregar los resultados a los clientes. Lo más importante en la entrega del servicio es la categoría y la prioridad asignada a la petición. Cuando se trata servicios de valor agregado, generalmente el trabajo es llevado a cabo por alguien que no es el agente que realizó el registro de la petición. Cuando son de soporte técnico, el agente que registra es usualmente el primero que trata de resolver el problema. Un aspecto sumamente importante en este proceso es realizar los trabajos en los tiempos predeterminados y con los costos más eficientemente posibles. Como se hace esto, depende en cómo se encuentre estructurada la organización.

2.3.2.5. Captura de Conocimiento.

La captura de conocimiento se usa para complementar las peticiones de soporte técnico. Este proceso pretende recoger y compartir el conocimiento colectivo de los agentes del centro de servicio (conocimiento institucional). Cuando uno identifica un nuevo problema y una solución, crea un expediente conocido como “reporte base de conocimientos”. Es entonces que el documento se hace disponible para todos los demás. Si un problema similar es reportado, el agente puede buscar por un reporte base. Esto elimina la necesidad de que se resuelva un problema que alguien más ya ha solucionado y documentado, lo que se traduce en un ahorro de tiempo.

Todos los problemas comunes y recurrentes deberían tener un reporte base asociado que detalle la solución aprobada por el centro de servicio. Realizar este proceso tiene tres beneficios principales. Primero, los agentes no pierden el tiempo intentando resolver algo ya está solventado. Segundo, todos los agentes aplican la solución estándar aprobada por el centro de servicio, lo que incrementa la consistencia. Finalmente, la realización de nuevos trabajos y nuevas soluciones son comunicados a todos los demás por medio de los reportes base.

2.3.3. Revisión de la Administración y Vigilancia.

El centro de servicio es un centro de intercambio de información para tanto problemas como peticiones de servicios. En este rol, el centro se encuentra en una posición única para recoger información sobre problemas en la estructura interna como problemas que los clientes tienen con los productos y servicios que la compañía ofrece. El centro también se encuentra en una posición única para recoger inteligencia sobre lo que los clientes, tanto los internos como los externos desean, en cuanto a mejoras en productos y servicios. La información es extremadamente valiosa para las operaciones internas, pues apunta a encontrar problemas en la infraestructura interna de la institución, lo que puede servir para implementar proyectos que eliminen los problemas que disminuyen la productividad.

La revisión y vigilancia también debe enfocarse internamente en el desempeño del centro de servicio. Este proceso crítico debe revisar y mejorar constantemente los procesos de administración de problemas, el personal, las herramientas y la estructura del centro. El objetivo es optimizar el desempeño continuamente, año con año.

2.3.4. Organización de un Centro de Servicio.

Cuando se organiza o reorganiza un centro de servicio, hay varios factores importantes a considerar. Pero ciertamente, el más importante es de éstos, es el tipo de centro que se planea colocar. Es necesario considerar si proporcionará el mejor servicio posible o simplemente el mínimo requerido para salir adelante. Se debe considerar el ambiente que actualmente existe y el ambiente al que se desea dar soporte en el futuro. También se debe considerar el volumen de peticiones que se esperan recibir, así como la complejidad de estas, para esto se puede usar la teoría de colas (Walker Gary, IT Problem Management, 2001) para determinar el número de recursos y las habilidades técnicas necesarias para manejar con anticipación el volumen y complejidad de las peticiones. Otro factor importante es el tipo de herramientas de soporte que se necesitan, las herramientas que se implementen en el centro de servicio afectan directamente el número de recursos y el modelo para organizar dichos recursos.

Basado en estos factores, existen varios enfoques que se pueden utilizar para maximizar la utilización de recursos para determinar los niveles de servicio y los objetivos, estos enfoques son referidos como “tipo de entrega de servicio” (Walker Gary, IT Problem Management, 2001). Los dos tipos son el inmediato y el gestionado. En el modelo inmediato, los clientes entran en la cola y esperan hasta que se encuentre disponible un agente, quien trata de resolver el problema al instante. En el modelo gestionado, el cliente generalmente envía un email o un fax, o llena un formulario Web, el cual entra en la cola. A diferencia del modelo inmediato, el modelo gestionado utiliza un controlador (manual o automático) que revisa,

prioriza, y luego asigna el ticket a un agente o grupo de agentes. Actualmente, muchos centros de servicio utilizan ambos modelos.

2.3.4.1. Modelo de Respuesta Inmediata.

Implementar este modelo significa que se intentarán manejar las peticiones en el momento en que llegan. Los clientes contactan al centro de servicio usando los medios que éste provee para ellos. Estos métodos pueden incluir telefonía, email, fax, etc. Una vez que el cliente contacta el centro, se debe tener algún mecanismo para encaminarlos hacia el agente, ingeniero o técnico disponible. Este mecanismo puede ser tan simple como una línea dedicada a cada agente, o muy sofisticado como sistema de distribución de llamadas que encamina al cliente a la cola más apropiada mediante los datos ingresados donde espera el próximo agente disponible. También existen mecanismos intermedios, como puede ser una o más recepcionistas que reciben las peticiones y luego manualmente las encaminan a la persona apropiada o cola en el siguiente nivel (tier).

En este modelo, el grupo o grupos de agentes que proveen respuesta inmediata se encuentran en el tier 1. El tier 1 puede ser un solo grupo de agentes o múltiples grupos de agentes. El modelo organizacional con un solo tier es el más simple de todos. “La teoría de colas establece que este modelo es el más eficiente para manejar el volumen de peticiones mientras se maximiza la utilización de recursos” (Walker Gary, IT Problem Management, 2001). Este modelo ignora la variabilidad y la complejidad, pues asume que cada recurso puede manejar cualquier problema que entre. Esto no es posible en cualquier centro de servicio, en ambientes que dan soporte a múltiples productos o plataformas no es razonable esperar que una persona pueda manejar todos los tipos de peticiones. Incluso si se pudieran encontrar recursos capaces de proveer ese nivel de servicio, sería mucho más costoso y menos eficiente utilizar los recursos expertos para manejar problemas recurrentes y muy simples.

Un ejemplo de un solo tier de un centro de servicio con respuesta inmediata se muestra en la *FIGURA 2.2*:

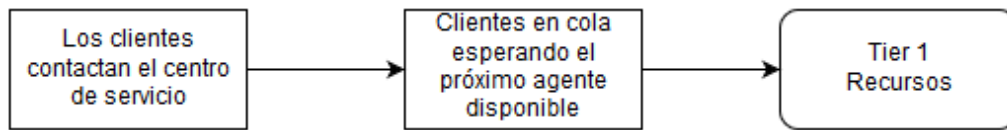


FIGURA 2.2: Modelo De Respuesta Inmediata
(Walker, Gary. IT Problem Management, 2001)

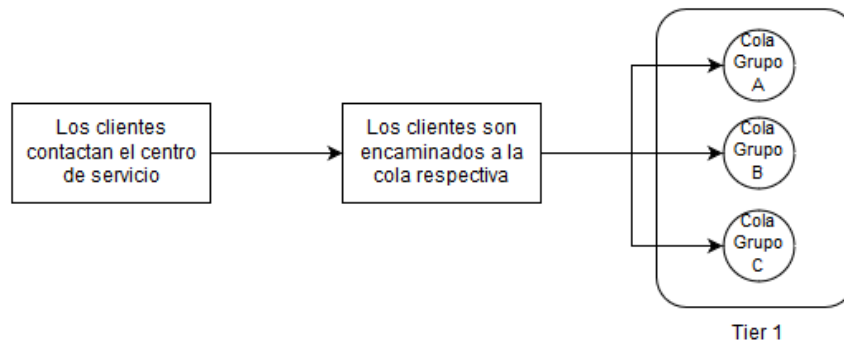


FIGURA 2.3: Modelo Respuesta Inmediata: Múltiples Grupos De Recursos
(Walker, Gary. IT Problem Management, 2001)

En la *FIGURA 2.3*, el tier se ha subdividido en múltiples grupos. Como se mencionó en párrafos anteriores, la teoría de colas establece que múltiples grupos son menos eficientes que uno solo. Esto toma sentido cuando se considera que los recursos en los grupos A y C podrían ser subutilizados, mientras los recursos en B podrían estar atendiendo a muchos clientes que esperan en esa cola. Esto no quiere decir que subdividir los recursos sea una idea mala. Por ejemplo, cuando se brinda soporte a múltiples productos o servicios sería irrazonable pensar que una sola persona puede saber todo de todos los productos, en este caso es necesario subdividir los recursos en grupos que sean especializados en uno o varios productos, o también en uno o varios tipos de problemas.

Hay tres factores que se necesitan considerar cuando se determinan el número de tiers que son requeridos y como organizar cada uno de ellos. Estos factores son la variabilidad, la complejidad y el volumen de las peticiones.

2.3.4.2. Variabilidad de las Peticiones.

La variabilidad se refiere a los diferentes tipos de peticiones que entran al centro de servicio, entre más tipos diferentes, más variabilidad. La variabilidad es importante para conocer la cantidad de conocimientos que se requiere para manejar las peticiones. Entre más grande la variabilidad implica que se necesitan más conocimientos para solventar las peticiones, entre menos variabilidad son necesarios menos conocimientos.

2.3.4.3. Complejidad de las Peticiones.

La complejidad y el volumen de las peticiones juegan un papel igual de importante en determinar la estructura organizacional de un centro de servicio. La complejidad se refiere al nivel de dificultad encontrado al darle mantenimiento a una petición. Muchas peticiones están bien definidas y son bastante conocidas para el personal. Este tipo de peticiones no son difíciles de resolver, por lo que son de baja complejidad. Otras peticiones ocurren con menor frecuencia y requieren más investigación. Éstas podrían involucrar interacción entre múltiples productos, funciones y características rara vez usadas, entre muchas otras cosas más. Problemas complejos toman más tiempo para ser resueltos y requieren más experiencia técnica que la que se encuentra disponible en los recursos de tier 1.

La complejidad de las peticiones debería ser usada para determinar el número de tiers de soporte que son necesarias. Por ejemplo, en un centro de servicio donde la mayoría de las peticiones son complejas podría bastar con un solo tier de múltiples expertos altamente capacitados, los cuales a su vez podrían resolver las peticiones simples que llegan. Caso contrario, cuando la mayoría de las peticiones son comunes, pero también llega cierto porcentaje de peticiones complejas, se necesitaría un tier para las simples y un segundo tier para resolver las peticiones

complejas que los recursos de tier 1 no pueden mantener debido a la dificultad o todo el tiempo que consumiría tratar de solucionarlas, tal como se muestra en la **FIGURA 2.4**.

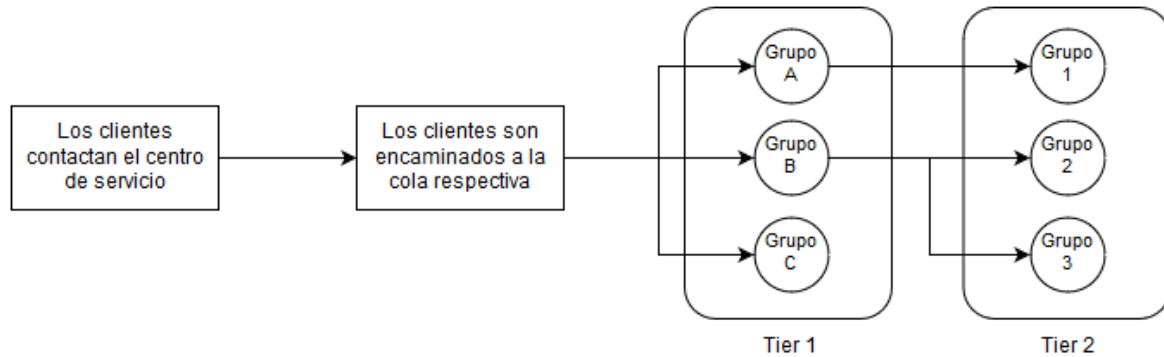


FIGURA 2.4: Modelo Multitier Con Múltiples Grupos
(Walker, Gary. IT Problem Management, 2001)

Como en el modelo de tier 1, la variabilidad ayudará a determinar cuántos grupos son necesarios en el segundo tier. En la **FIGURA 2.4**, el grupo B en el tier 1 puede escalar las llamadas a diferentes grupos. Esto es porque cada grupo del tier 2 podría atender diferentes tipos de peticiones complejas según lo defina la variabilidad.

Hay algunas desventajas en usar un Modelo Multitier, entre estas tenemos que es mucho más difícil de gestionar y requiere niveles extra de administración. Las peticiones toman más tiempo en ser resueltas a medida que avanzan en los niveles, puesto que son agregadas a una nueva cola. Usualmente cuando un cliente es desplazado a un nuevo nivel tiene que volver a explicar su petición y contestar las preguntas que esto conlleva.

2.3.4.4. Volumen de las Peticiones.

Cuando se usa la complejidad de las peticiones como un medidor que determina la necesidad de otro nivel de soporte, también debe considerarse el volumen de peticiones complejas. Por ejemplo, si todas las peticiones son simples y

pueden ser gestionadas por los recursos del tier 1, probablemente no sea necesario añadir un segundo nivel. Como se ha mencionado, agregar más niveles también agrega complejidad y sobrecarga adicional al centro de servicio, aparte que también agrega retrasos en el tiempo de resolución. Bajo este escenario, una mejor solución sería agregar recursos más altamente capacitados en el tier 1. Otro enfoque sería utilizar estos recursos mejor capacitados como mentores de todos los demás agentes del nivel. Bajo este rol, ellos solo tomarían peticiones después que los agentes hayan intentado resolver el problema, o durante situaciones de sobre flujo de peticiones.

2.3.5. Modelo de Respuesta Gestionada.

Desde un punto de vista organizacional, este modelo es muy similar al modelo de respuesta inmediata. Las diferencias son las siguientes:

- Los clientes pueden usar métodos alternativos para contactar al centro de servicio.
- Se tiene la oportunidad de gestionar la asignación y la terminación de las peticiones para balancear eficientemente la carga de trabajo.
- Se tiene la oportunidad de priorizar las peticiones y gestionarlas en un orden diferente que en el modelo de respuesta inmediata.



FIGURA 2.5: Modelo de respuesta gestionada
(Walker, Gary. IT Problem Management, 2001)

Como se puede observar en la *FIGURA 2.5*, la organización de los recursos en este modelo no es diferente a la estructura del modelo de respuesta inmediata. El mismo personal puede manejar las peticiones. Y al igual que en el modelo de respuesta inmediata, como las peticiones son encaminadas a la respectiva cola depende de las herramientas que el centro de servicio posea. La principal ventaja de este modelo es que permite distribuir la carga de trabajo y de esta manera suavizar las altas y bajas en el volumen de las peticiones. También permite asignar peticiones a la persona más apropiada para un determinado problema.

Este modelo también tiene desventajas entre las que se pueden mencionar que retrasa la solución de algunas peticiones, lo que no es aceptable para algunos clientes. Otra desventaja más significativa es que tiende a generar más contacto entre el centro de servicio y el cliente. Esto ocurre porque el agente que va a manejar la petición a menudo necesita información adicional por lo que necesita contactar al cliente. Si el cliente no se encuentra disponible, la petición se irá retrasando, y lo mismo ocurre cuando el cliente decide contactar al agente, puede que éste se encuentre resolviendo otras peticiones.

2.3.6. Descripción de la situación actual en el Área de Soporte Técnico y Tecnología Informática.

Aplicaremos el modelo planteado anteriormente como base para la descripción de la situación actual de los procesos que se llevan a cabo en el Centro de Servicios Interno, en este caso el Área de Soporte y Tecnología Informática de la Facultad Multidisciplinaria de Occidente de la Universidad de El Salvador, para determinar la administración del trabajo que realiza.

El Área no solo trabaja como un Centro de Soporte Técnico, también se especializa en brindar mantenimiento informático a distintos tipos de equipos, por lo que la dividiremos en base a estos objetivos como un Centro Servicios Interno y como un Centro de Mantenimiento Informático.

2.3.6.1. Como un Centro de Servicios Interno.

El soporte técnico que el Área proporciona a los usuarios internos tiene como objetivo la solución de problemas que se presenten en determinados Departamentos y Unidades, pero si se presenta la situación ésta brindará su servicio a empleados y sectores fuera de su régimen. El servicio que se brinda es presencial, pero la comunicación con el Área se puede realizar en persona, así como por teléfono. El solicitante nunca forma parte del proceso de solución.

Para mejor comprensión del servicio brindado por el Área, lo dividiremos en 3 procesos generales los cuales son: solicitud, identificación del problema y solución.

2.3.6.2. Solicitud.

El proceso de solicitud ocurre cuando el usuario no puede resolver un problema que tiene el equipo informático asignado o compartido, como lo son las impresoras. Al quedarse sin opciones, éste recurre a realizar una solicitud al Área, ya sea por teléfono o en persona en dado caso que no se puedan contactar con el personal pertinente.

Por lo general, el solicitante llena un formulario donde detalla lo que cree es el problema que se está presentando con el equipo informático. En caso de no contar con el formulario, un técnico en el Área se encargará del registro y se procederá con la identificación del problema.

2.3.6.3. Identificación del problema.

Se asigna un técnico que se encargará de realizar la asistencia de forma personal al solicitante. El primer paso es escuchar el problema que el usuario está experimentando al utilizar el equipo informático, de esta forma pueden identificar la posible causa, específicamente si se trata del hardware o software.

Al identificar la causa del problema, el técnico toma la decisión de realizar una labor de mantenimiento en el lugar, o un servicio de configuración, que es común

cuando se trata de un problema de software, sí es más compleja y requiere de mayor atención se procede con la desconexión del equipo y se traslada al Área, para un diagnóstico más profundo y la realización de un mantenimiento correctivo.

2.3.6.4. Solución.

Sí se trata servicio de configuración o mantenimiento en el lugar, el técnico asignado se encarga de solventar el problema, ya que el solicitante no forma parte del proceso de solución, se detalla el por qué se generó tal situación, explicando paso a paso cómo se solucionó y se comprueba que todo funcione con normalidad. La idea es dejar una noción más clara al usuario para que en un futuro le ayude a solventar de una manera más rápida el problema si éste ocurre de nuevo.

En caso de que el equipo informático haya sido trasladado al Área, al finalizar el mantenimiento correctivo, se traslada el equipo de nuevo y se conecta, corroborando que funcione hardware y software perfectamente. Se notifica al solicitante cuál fue el problema, la solución y se dan recomendaciones para evitar que la causa no se presente nuevamente.

En todos los casos, se pide que proporcionen una firma en el mismo formulario de solicitud como comprobante del servicio realizado.

2.3.7. Como un Centro de Mantenimiento Informático.

El Área se encarga de brindar soporte técnico a distintos sectores de la Facultad, pero cuando se detecta un problema en un equipo informático, son los encargados de brindar mantenimiento correctivo para la solución de dicho problema. No obstante, también se encarga de evitar que el equipo informativo tenga problemas de hardware o de software, implementando un plan de mantenimiento preventivo.

Ya que el mantenimiento correctivo está ligado al proceso de soporte técnico, el mantenimiento preventivo se realiza a cada equipo de cómputo de cada en base a

un cronograma estipulado por el Área de Soporte y Tecnología Informática. A continuación, explicaremos cómo se lleva a cabo cada proceso.

2.3.7.1. Mantenimiento Correctivo.

En este proceso el equipo es revisado partiendo de la descripción dada por el usuario solicitante, para determinar el fallo, se hacen pruebas en el hardware y revisiones de software, cuando se ha determinado el problema se procede a la reparación la cual puede ser desde una simple actualización de software a la reinstalación del Sistema Operativo, donde se advierte al solicitante que se realizará un back-up de la información almacenada en el equipo, o en caso de ser un problema en una pieza de hardware se realizará un cambio, el tiempo que el equipo tarda en ser entregado al solicitante, varía debido a la carga de trabajo que ya se tenía con anterioridad y la prioridad del caso.

Sin importar cuál sea el problema del equipo se realiza una limpieza general al hardware y al software, como una buena práctica y extender su vida útil.

2.3.7.2. Correspondencia.

Este proceso se realiza antes de brindar cualquier mantenimiento preventivo. Se notifica días antes al Departamento o Área en el que se llevará a cabo el mantenimiento preventivo, anexando así un cronograma que detalla la identificación del equipo informático, su encargado y que día se realizará.

2.3.7.3. Mantenimiento Preventivo.

Al igual que el mantenimiento correctivo, el equipo es trasladado al Área para desmontar la mayoría de las piezas de hardware. Se procede a realizar una limpieza profunda de cada periférico, monitor, UPS y pieza de hardware, luego de finalizada se vuelven a montar a su lugar respectivamente.

Después de la limpieza y verificar que todas las piezas funcionen correctamente, se realiza una revisión de software, se actualizan los programas a su

última versión y se hace un análisis con el antivirus. En caso de que se encuentre una gran deficiencia en el software que la configuración no pueda solventar, se llevará a cabo una reinstalación del Sistema Operativo, al igual como podría ocurrir en el mantenimiento correctivo, se crea un back-up de los datos de los usuarios del equipo y se procede con la reinstalación. Se instalan los utilitarios más importantes y se ubica el Back-up en el escritorio.

Cuando todo funciona correctamente, el equipo de cómputo es trasladado al lugar donde pertenece, se conecta, y de nuevo se verifica que todo funcione correctamente.

2.3.8. Administración de registros.

El Área se encarga de llevar un registro por cada solicitud, mantenimiento correctivo y preventivo de cada equipo informático.

Todos estos registros se administran en bitácoras, cada una asociada a un equipo informático, la cual contiene la información de cuantos mantenimientos ha recibido.

Capítulo III: Diseño Metodológico

3.1. Introducción.

En este capítulo se da a conocer el enfoque de la investigación como una guía más clara de la realización de dicho proceso; se describe el tipo de investigación realizada para establecer la propuesta, las diferentes fuentes de información, métodos y técnicas que se han utilizado para su puesta en marcha.

3.2. Enfoque de la Investigación.

Debido a la naturaleza de la investigación, es realizada bajo un enfoque cualitativo (Hernández, Fernández y Baptista, 2010), ya que los datos analizados sin medición numérica se interpretan para descubrir o afinar preguntas de la investigación. La comprensión de los datos proporcionados es de vital importancia en ambientes de desarrollo.

Mediante el enfoque cualitativo se plantean los siguientes propósitos de la investigación:

- El planteamiento de cada objetivo por separado.
- Enfocarse en explorar y comprender un solo concepto o idea de los datos analizados que han sido proporcionado como una visión aproximada del software final que cumpla las funciones deseadas, aunque comúnmente en el enfoque cualitativo es posible identificar y analizar relaciones en otros conceptos conforme se desarrolla el software ya que no es posible identificarlos cuando se inicia el proyecto.
- Identificar la infraestructura o ambiente inicial para el desarrollo del software.
- El diseño de la Base de Datos y elaboración del Diccionario de Datos.
- Elaboración de Manuales de Usuario.

3.3. Tipo de Investigación.

La investigación realizada es de tipo tecnológica, pues consiste en la implementación de un sistema informático en el Área de Soporte y Tecnología de la

Facultad Multidisciplinaria de Occidente mediante la aplicación de técnicas de ingeniería, para lo cual es necesario realizar un diagnóstico de las necesidades actuales, establecer el diseño de la solución, ejecutar el desarrollo de dicho diseño e implementarlo en la institución evaluando que solventa los problemas y satisface las necesidades del Área. (Cegarra Sánchez, Metodología de la Investigación Científica y Tecnológica, 2004).

3.4. Diseño de Investigación.

La finalidad de la investigación-acción, como uno de los tipos de diseño de la investigación cualitativa, es resolver problemas “cotidianos e inmediatos” y mejorar prácticas concretas. Su propósito fundamental se centra en aportar información que guíe la toma de decisiones para programas, procesos y reformas estructurales (Álvarez Gayou, 2003; Merriam, 2009). Aunque en principio se ideó para resolver problemas de carácter social, también puede ser adaptado a diferentes investigaciones cualitativas que cumplan con sus características.

La investigación-acción construye el conocimiento por medio de la práctica. Entre las principales características de este diseño están (Sandín, 2003):

1. La investigación-acción envuelve la transformación y mejora de una realidad (social, educativa, administrativa, etc.).

Para esta investigación, es necesario automatizar los procesos relacionados con el registro de información que actualmente se llevan a cabo en el Área, lo que se puede traducir en un mejoramiento de la realidad.

2. Parte de problemas prácticos y vinculados con un ambiente o entorno.

El entorno o ambiente es el Área de Soporte y Tecnología y los problemas prácticos son todos los factores que afectan su forma de trabajo.

3. Implica la total colaboración de los participantes en la detección de necesidades, ya que ellos conocen mejor que nadie la problemática a

resolver, la estructura a modificar, el proceso a mejorar y las prácticas que requieren transformación y en la implementación de los resultados del estudio.

Para conocer a profundidad los requerimientos del sistema y qué problemas afectan al Área es indispensable la colaboración directa con el personal involucrado y para realizar la evaluación del sistema.

Para complementar el diseño, se optará por la perspectiva técnico-científico, de la investigación cualitativa, la cual consiste en un conjunto de decisiones en espiral las cuales se basan en ciclos repetidos de análisis para conceptualizar y redefinir el problema las veces que sean necesarias. Así, la investigación-acción se integra con fases secuenciales de acción: planificación, identificación de hechos, análisis, implementación y evaluación (Álvarez-Gayou ,2003).

Al contrastar dichas fases con el ciclo de vida en el desarrollo de un sistema, se observa que encajan perfectamente desde la recolección de requerimientos hasta la implementación y la posterior retroalimentación para la mejora continua del sistema y corrección de errores.

Metodología de la Investigación.

La metodología de la investigación tecnológica se ha adaptado para las carreras de ingeniería tomando como base las propuestas metodológicas de García Córdoba (La perspectiva de la investigación tecnológica en la Educación, 2012) y Rincón (Investigación científica y tecnológica como factores de innovación, 2012).

Las etapas por seguir son las siguientes:

1. Planteamiento del Estudio:
 - a. Planteamiento del problema. El problema de investigación parte de una necesidad.
 - b. Formulación del problema. La formulación del problema también puede ser en forma de lista de requerimientos y restricciones o en forma de pregunta.

- c. Objetivos.
 - d. Justificación e importancia.
2. Marco Teórico:
 - a. Antecedentes del problema.
 - b. Bases teóricas.
 - c. Definición de términos básicos.
 3. Metodología del Diseño:
 - a. Plan de trabajo del diseño: Pasos a seguir para alcanzar los objetivos.
 - b. Recursos. Instrumentos, herramientas, materiales e inversiones.
 4. Desarrollo e Implementación: En esta etapa se lleva a cabo el desarrollo del diseño.
 5. Evaluación: Se evalúa si el diseño realizado cumple con todos los requerimientos o restricciones establecidas al inicio de la investigación mediante pruebas o simulaciones.

En la investigación tecnológica no es posible establecer variables iniciales de estudio, pues su finalidad es "transformar la realidad" (García Córdoba, La perspectiva de la investigación tecnológica en la Educación, 2012), es decir, que las variables cambiarán a medida que evoluciona el diseño o desarrollo.

3.5. Recursos y Herramientas de la Investigación.

En esta sección se describirán cada uno de los recursos, herramientas, técnicas y metodologías utilizadas en el planteamiento, análisis e interpretación del problema como en el desarrollo del sistema.

3.5.1. Técnicas de Diagnóstico y Formulación del Problema.

El diagnóstico se refiere principalmente a la identificación y estudio de aquellos elementos que se desean modificar o en los que se ha detectado una situación indeseable. La formulación del problema consiste en la definición de elementos que generan insatisfacción, que impiden o dificultan lograr un objetivo.

3.5.1.1. Diagrama Causa-Efecto.

Es una herramienta efectiva para estudiar procesos y situaciones, y para desarrollar un plan de recolección de datos.

El Diagrama de Causa y Efecto es utilizado para identificar las posibles causas de un problema específico. La naturaleza gráfica del Diagrama permite que los grupos organicen grandes cantidades de información sobre el problema y determinar exactamente las posibles causas. Finalmente, aumenta la probabilidad de identificar las causas principales.

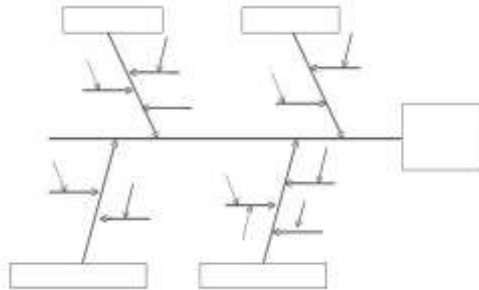


FIGURA 3.1: Diagrama de Ishikawa
(Jones, Christopher, Métodos de Diseño)

3.5.1.2. Diagrama de Caja Negra.

Es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software.

En las pruebas de caja negra, se enfoca solamente en las entradas y salidas del sistema, sin preocuparse en tener conocimiento de la estructura interna del proceso. Para obtener el detalle de cuáles deben ser esas entradas y salidas, nos basamos únicamente en los requerimientos de software y especificaciones funcionales.

Este permitirá tener una visión general del problema a resolver, el cual parte de un estado “A” (situación actual) hasta un estado “B” (situación deseada); pasando

a través de un proceso de transformación para obtener una solución. Su representación gráfica se muestra en la *FIGURA 3.1*.



FIGURA 3.2: Método de la Caja Negra

(Jones, Christopher, Métodos de Diseño, Pág. 42)

3.5.2. Técnicas de Análisis y Diseño.

Para la realización del análisis de requerimientos y el diseño del sistema informático se han de utilizar la siguiente técnica:

3.5.2.1. UML 2.0

Se definirá la especificación de requerimientos y el diseño del sistema por medio del lenguaje de modelado UML (Lenguaje Unificado de Modelado), que es un lenguaje que permite modelar, visualizar, especificar, construir y documentar los artefactos de un sistema, desde una perspectiva orientada a objetos (López, P. H. Ciclo II - 2010). Constituye el lenguaje de modelado estándar más utilizado para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

UML proporciona varios tipos de diagramas a través de una notación específica, que cuando se utilizan dentro de una metodología dada, aumentan la facilidad de comprensión de una aplicación en desarrollo. Las razones principales para la utilización de UML son las siguientes:

- UML proporciona un lenguaje común para los analistas de negocios y desarrolladores, esto permite la comunicación y comprensión.
- UML permite modelar visualmente cómo opera un negocio. Quién, Cómo y Qué actividad realiza, está representado en términos de símbolos y diagramas.

- UML es utilizado para diagramar software desde una perspectiva orientada a objetos. Los objetos que componen los modelos UML, reflejan mejor las entidades del mundo real, por lo que son más flexibles e intuitivos.
- UML tiene muchos tipos de diagramas que permiten representar la información desde diferentes puntos de vista.
- UML ayuda a enfocarse en el cliente, debido a que gira en torno a los casos de uso de negocio, que muestran cómo un proceso de negocio entrega valor al cliente.
- UML ayuda a obtener requisitos del sistema a través de las descripciones de las relaciones entre los componentes, los actores y otras entidades.

El resultado final de la aplicación de UML, es el modelado de la lógica y los procesos del negocio, que resuelve un problema de negocio específico, responde a las necesidades reales de la organización, y ofrece un valor agregado a los clientes.

Por su nivel de importancia y el apoyo que dan en la representación de los requerimientos y los procesos de negocio del proyecto a desarrollar, se utilizarán los siguientes diagramas:

- Diagramas de Comportamientos
 - Diagrama de Casos de Uso
- Diagramas Estructurales
 - Diagrama de Clases

3.5.2.2. Diagrama de Casos de Uso.

El diagrama de casos de uso actúa como foco en la descripción de los requisitos del usuario. En él se describen las relaciones entre los requisitos, los usuarios y los componentes primarios. En los diagramas de casos de uso se resume quién utiliza la aplicación o sistema y qué puede hacer con ella. A continuación, se presenta un bosquejo del diagrama de casos de uso con su respectivo significado.

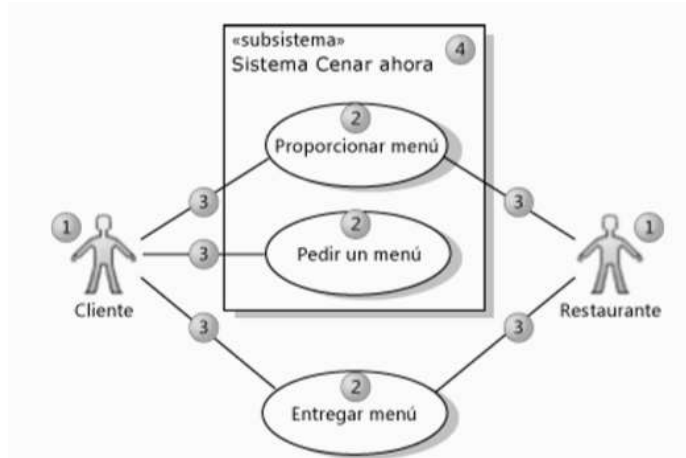


FIGURA 3.3: Diagrama de Caso de Uso
 (Diagramas de casos de uso de UML: Instrucciones. Microsoft. Recuperado de <https://msdn.microsoft.com/library/dd409432.aspx>)

N°	Elemento	Descripción
1	Actor	Representa un usuario, organización o sistema externo que interactúa con la aplicación o sistema.
2	Caso de Uso	Representa las acciones que realizan uno o varios actores para conseguir un objetivo determinado.
3	Asociación	Indica que un actor forma parte de un caso de uso.
4	Subsistema	Sistema o aplicación (o alguno de sus componentes) en el que se está trabajando. Los casos de uso que un sistema o componente admiten aparecen dentro de su rectángulo.

TABLA 3.1: Elementos de Diagrama de Caso de Uso
 (Diagramas de casos de uso de UML: Instrucciones. Microsoft. Recuperado de <https://msdn.microsoft.com/library/dd409432.aspx>)

3.5.2.3. Diagrama de Clase.

En este diagrama se describe el objeto y las estructuras de información que se utilizan en una aplicación, tanto de forma interna como en la comunicación con los usuarios. Esta información se describe sin hacer referencia a ninguna implementación concreta. Las clases y relaciones se pueden implementar de muchas maneras, por ejemplo, en tablas de bases de datos, en nodos XML o en composiciones de objetos de software. A continuación, se presenta un bosquejo del diagrama de clases con su respectivo significado.

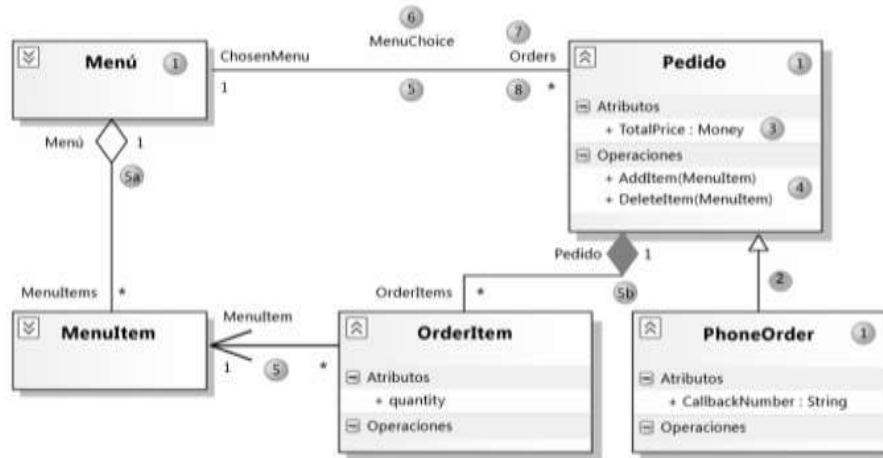


FIGURA 3.4: Diagrama de Clase
 (Diagramas de clases UML: Referencia. Microsoft. Recuperado de <https://msdn.microsoft.com/library/dd409437%28v=vs.120%29.aspx>)

N°	Elemento	Descripción
1	Clasificador	Nombre general de una clase, interfaz o enumeración. Los componentes, casos de uso y actores también son clasificadores.
1	Clase	Nombre general de una clase, interfaz o enumeración. Los componentes, casos de uso y actores también son clasificadores.
2	Generalización	El clasificador específico hereda parte de su definición del clasificador general. El clasificador general se encuentra en el extremo del conector de la flecha. El clasificador específico hereda los atributos, las asociaciones y las operaciones.
3	Atributo	Valor con tipo que se adjunta a cada instancia de un clasificador.
4	Operación	Método o función que pueden realizar las instancias de un clasificador.
5	Asociación	Relación entre los miembros de dos clasificadores.
5a	Agregación	Asociación que representa una relación de propiedad compartida.
5b	Composición	Asociación que representa una relación parte/todo.
6	Nombre de Asociación	Nombre de una Asociación. El nombre puede quedarse vacío.
7	Nombre de Rol	Nombre de un rol, es decir, un extremo de una asociación. Puede utilizarse para hacer referencia al objeto asociado.
8	Multiplicidad	Indica cuántos de los objetos de este extremo se pueden vincular a cada objeto del otro.

TABLA 3.2: Elementos de Diagrama de Clase
 (Diagramas de clases UML: Referencia. Microsoft. Recuperado de <https://msdn.microsoft.com/library/dd409437%28v=vs.120%29.aspx>)

3.5.3. Metodología de Diseño de Bases de Datos.

Una base de datos es una colección de datos relacionados que representan algún aspecto del mundo real (Elmasri, Ramez B., & Navathe, S. B. 2007). El buen diseño de la base de datos es clave en el desarrollo del sistema informático, debido a que un sistema bien diseñado es fácil de desarrollar y mantener, mientras que los sistemas mal diseñados son costosos de mantener, difíciles de probar y poco fiables.

Para lograr un buen diseño de base de datos se utilizarán modelos de datos que permitan representar diferentes niveles de abstracción (conceptual, lógico y físico) de la base de datos.

Los modelos de datos son una colección de herramientas conceptuales que permiten describir y comprender datos, las relaciones entre estos, restricciones de consistencia, entre otros (Silberschatz, A., Korth, Henry F., & Sudarsha, S. 2002).

A continuación, se describen los modelos de datos utilizados en el diseño de la base de datos:

3.5.3.1. Modelo de Datos Conceptual.

Es un Modelo orientado a la descripción de estructuras de datos y restricciones de integridad, en este modelo se incluye descripciones detalladas de los tipos de entidades, atributos, relaciones y restricciones. Para poder representar el Modelo de Datos Conceptual se utilizará el Modelo Entidad Relación Extendido (EER).

3.5.3.2. Modelo Entidad Relación (ER)

Es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información, así como sus interrelaciones y propiedades. Su representación gráfica se muestra en la *FIGURA 3.4*.

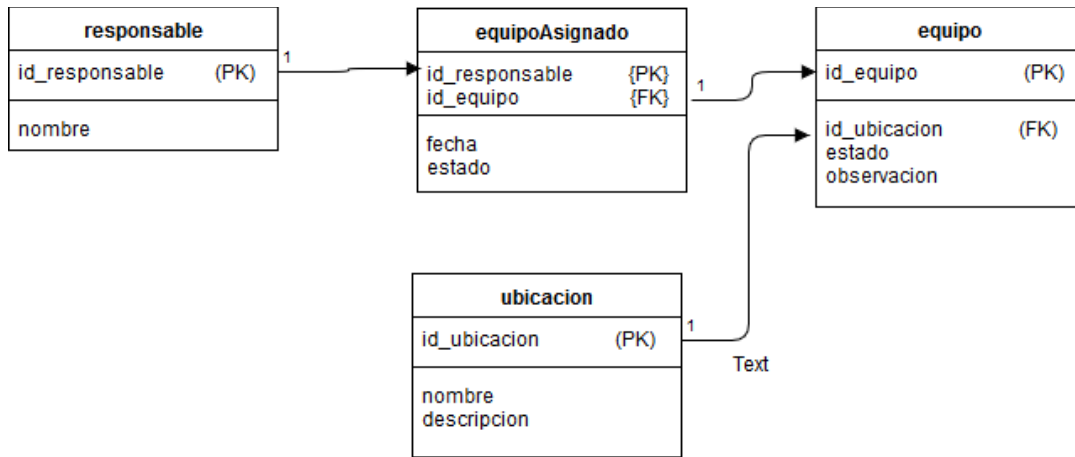


FIGURA 3.5: Bosquejo Diagrama Entidad Relación (Fuente propia)

3.5.4. Descripción de Herramientas.

Además de la aproximación metodológica, uno de los factores determinantes en el éxito de un proyecto informático, lo constituye la selección de herramientas de trabajo que se utilizarán durante su desarrollo. Por tal razón, se han seleccionado la más conveniente según determinados factores.

A continuación, se describen las herramientas a utilizar en el desarrollo del proyecto:

Herramienta	Descripción
Google Drive	Es un servicio de almacenamiento de archivos en línea. Será utilizado para el almacenamiento de archivos y documentos correspondientes al proyecto.
Navegadores Web	Es una herramienta que opera a través de internet, interpretando la información de archivos y sitios web para que estos puedan ser leídos.
Microsoft Word (Online)	Programa informático orientado al procesamiento de texto.
Microsoft Excel (Online)	Programa informático orientado al procesamiento de hojas de cálculo.
Draw.io	Sitio web para la elaboración de diagramas.

TABLA 3.3: Herramientas

3.6. Técnicas de Recolección de Datos.

Para realizar el proceso de recolección de datos se ha de utilizar la siguiente técnica:

3.6.1. Observación.

La recolección de datos resulta fundamental en cualquier investigación, utilizando el enfoque cualitativo se busca obtener información de personas, contextos o situaciones en profundidad con la finalidad de analizarla y comprenderla para responder a las preguntas de investigación.

La observación cualitativa implica adentrarse en profundidad a la situación estudiada y mantener un papel activo y de reflexión permanente, es necesario estar atento a los detalles, sucesos, eventos e interacciones.

Los propósitos esenciales de la observación aplicados a esta investigación son:

- Describir el contexto o ambiente y las actividades que se desarrollan en éste.
- Comprender los procesos, vinculaciones entre personas y los eventos que suceden a través del tiempo.
- Identificar problemas.

Los elementos específicos que se han de observar son:

- Ambiente físico.
- Actividades que realizan los participantes.
- Artefactos utilizados.
- Hechos relevantes

En la observación cualitativa regularmente no se dispone de formatos o formularios estandarizados al inicio de la investigación, por este motivo, se debe

registrar todo lo que se considere importante. Conforme avanza la inducción, se puede generar listados de elementos de lo que debe ser analizado y obtener así modelos o guías de observación.

3.7. Cronograma de Actividades

Cronograma de Actividades																														
Actividades	Años																													
	2018												2019												2020					
	Meses												Meses												Meses					
	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic	Ene	Feb	Mar	Abr	May	Jun	Jul
1	Anteproyecto	x																												
Primera Fase:																														
2	Presentación de Anteproyecto		x																											
3	Planteamiento del Problema		x	x	x																									
4	Elaboración del Marco Teórico					x																								
5	Elaboración del Diseño Metodológico					x	x																							
6	Primera Defensa						x																							
Segunda Fase:																														
7	Diseño del Sistema						x	x																						
8	Desarrollo del Sistema							x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
9	Evaluación de Sistema																								x	x	x			
10	Ajustes y Correcciones																									x	x	x	x	x
11	Defensa Publica																												x	

3.8. Presupuesto del Proyecto.

Para la elaboración del presupuesto se ha escogido utilizar un cálculo por proyecto, por lo que los tiempos presentados equivalen al total de horas-hombre trabajadas las cuales en su defecto no necesariamente fueron de manera consecutiva.

Recurso Humano	Cantidad	Salario Mensual	Meses Laborados	Total a Pagar
Programador Java Junior	4	\$500.00	3	\$6,000.00
Subtotal				\$6,000.00

TABLA 3.4: Presupuesto de salarios del personal

Material Gastable	Cantidad	Costo Unitario	Costo Total
Impresiones	150	\$0.10	\$15.00
Lápices	4	\$0.20	\$0.80
Lapiceros	4	\$0.25	\$1.00
Folder	6	\$0.30	\$1.80
Memoria USB 16GB	2	\$8.00	\$16.00
Subtotal			\$34.60

TABLA 3.5: Presupuesto de materiales gastables en la investigación

Equipo	Cantidad	Precio Unitario	Precio Total
Cybertron PC	1	\$630.00	\$630.00
Lenovo X230	1	\$299.00	\$299.00
Lenovo G580	1	\$499.00	\$499.00
Dell Inspiron 5000 series	1	\$1200.00	\$1200.00
Samsung J5 Prime (HotSpot)	1	\$207.00	\$207.00
Subtotal			\$2,825.00

TABLA 3.6: Características de los equipos utilizados en la investigación

Gastos Directos	Meses	Cantidad Mensual	Total
Combustible	3	\$3.75	\$11.25
Alimentación	3	\$5.00	\$15.00
Viáticos	3	\$21.00	\$63.00
Energía eléctrica	3	\$10.00	\$30.00
Internet Residencial	3	\$25.00	\$75.00
Datos móviles	3	\$10.00	\$30.00
Subtotal			\$224.25

TABLA 3.7: Presupuesto de gastos directos en la investigación

Concepto	Gasto
Material Gastable	\$34.60
Equipo	\$2,825.00
Gastos Directos	\$224.25
Recurso Humano	\$6,000.00
Total	\$9,083.85

TABLA 3.8: Presupuesto total de la investigación

Capítulo IV: Desarrollo

4.1. Especificación de requerimientos de software.

La Especificación de Requerimientos de Software es una descripción completa del comportamiento del sistema que se va a desarrollar. Para la especificación de requerimientos del Sistema para el Área de Soporte y Tecnología informática se implementarán las prácticas recomendadas en el estándar IEEE 830-1998, Recommended Practice for Software Requirements Specifications.

Además de la perspectiva del producto, se presenta la especificación de los requerimientos categorizados en interfaces externas, requerimientos funcionales y otros tipos de requerimientos.

4.1.1. Introducción.

4.1.1.1. Propósito.

Especificar y clasificar de manera formal y oficial, los requerimientos funcionales, interfaces, restricciones de diseño y desarrollo del Sistema Informático que se ha de desarrollar e implementar. El documento de especificación de requerimientos de software está sujeto sujeta al estándar IEEE 830-1998 y cambios en los requerimientos, por las negociaciones de las partes del equipo de desarrollo y los interesados, generando nuevas versiones hasta su aprobación, una vez aprobado, servirá de base para las etapas siguientes del desarrollo del Sistema Informático.

4.1.1.2. Visión general.

La segunda sección de este documento, Descripción general (ver apartado 4.1.2.), brinda una descripción de la funcionalidad del Sistema, en la que se describe de forma general, cual es la perspectiva del producto a desarrollar, la funcionalidad que presentará, las características de los usuarios involucrados y todas aquellas restricciones que intervienen en el desarrollo del sistema. Esto presenta un contexto o marco de referencia para la especificación de requerimientos del sistema.

La tercera sección, Especificación de requerimientos (ver apartado 4.1.3.), describe en términos técnicos y con mayor profundidad de detalle los requerimientos del sistema.

4.1.2. Descripción general.

4.1.2.1. Perspectiva del producto.

La Perspectiva del producto consiste en presentar una visión y análisis del producto a desarrollar, que permita al lector comprender cómo los elementos del producto da soporte a la lógica del negocio. A continuación, se presenta la perspectiva del producto.

Será un sistema multiplataforma que funcionará en un ambiente web (ver apartado 4.2), por lo tanto, los usuarios podrán acceder al sistema por medio de un navegador web que les permitirá interactuar con los elementos de éste, desarrollado bajo el Modelo de Tres Capas, el cual se describe a continuación:

- Capa de presentación: Capa con la que el usuario interactúa, denominada también capa de usuario, debido a que presenta al usuario la información solicitada y captura los datos brindados por éste. Esta capa se comunica únicamente con la capa de negocio.
- Capa de negocio: Se denomina también capa de lógica del negocio porque es aquí donde se establecen todas las reglas del negocio que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes o peticiones de usuario y presentar los resultados o respuestas a dichas solicitudes, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.
- Capa de datos: Es donde residen los datos, recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

El Sistema será una herramienta utilizada por el Área de Soporte y Tecnología Informática para los procesos de gestión de la información, así como los usuarios que desean solicitar un servicio de soporte.

4.1.2.2. Funciones del producto.

Las Funciones del producto son el conjunto de características que permiten la realización de tareas que satisfacen las necesidades de los usuarios. El Sistema es desarrollado con el objetivo de cubrir y dar soporte a ciertas necesidades de los usuarios en los procesos de mantenimiento, por lo cual brindará las siguientes funcionalidades:

- Administración de elementos del sistema: Se conoce como elementos del sistema a cada uno de los elementos involucrados en los procesos de negocio, ya sea directa o indirectamente. Los elementos pueden ser: Unidades y Departamentos, docentes, técnicos, equipo informático, etc. El sistema proveerá de las interfaces de usuario necesarias para la gestión de cada uno de los elementos.
- Gestión de solicitudes de servicio de mantenimiento: Su principal función es la creación y seguimiento de las solicitudes de mantenimiento, es decir, mostrar al usuario el estado en que se encuentra la solicitud realizada, lo que ayudará al área de mantenimiento a una mejor organización de su trabajo, permitiendo con esto, dar una respuesta más rápida.
- Plan de mantenimiento preventivo: El sistema brinda la función de almacenar la programación anual de los mantenimientos preventivos del equipo que lo requiera, lo que permitirá visualizar la asignación de los periodos de tiempo en que se han de realizar los mantenimientos.
- Generación de reportes, estadísticas y bitácoras: El sistema permitirá la generación de reportes de ciertos apartados necesarios para los procedimientos administrativos que son realizados en el área de mantenimiento.
- Gestión de equipos: La aplicación tendrá la funcionalidad de agregar, modificar y consultar equipo informático a los que se les ha realizado un mantenimiento y observar información de importancia de cada uno.

4.1.2.3. Características de los usuarios.

Esta subsección describe las características generales de los usuarios del producto, incluyendo nivel educacional y experiencia técnica.

Los actores o personas que intervienen en el sistema se clasificarán en categorías diferentes, lo que permite una mejor comprensión de las diferentes responsabilidades (roles) que tendrá cada usuario dentro del sistema. Los dos diferentes roles en los que se categorizan son:

- Administrador general del Sistema: Usuario que es jefe del Área de Soporte y Tecnología Informática, tiene a su cargo un conjunto de dependencias a las que les brinda los servicios de soporte y personal técnico para la realización de servicios de mantenimiento correctivo y preventivo. Tendrá total control sobre cada uno de los elementos que el Sistema ofrece.
- Técnico del Área: Usuario que forma parte del área o departamento de mantenimiento. Este usuario es asignado a una solicitud y se encarga de solventarla.
- Solicitante: Usuario que es docente o personal administrativo, tiene a su cargo equipos informáticos pertenecientes a la Facultad. Este usuario podrá realizar solicitudes de mantenimiento de los recursos bajo su cargo.

Para una mejor comprensión de cada uno de los roles, en las siguientes tablas (*TABLA 4.1*, *TABLA 4.2* y *TABLA 4.3*) se presentan los perfiles y un conjunto de características ideales que los usuarios deben poseer, para asumir y desempeñar de la mejor manera un determinado rol dentro del sistema:

Rol	Administrador general del sistema	Código	adminGS
Nivel Académico			
Ingeniero de Sistemas Informáticos, Computación o Áreas afines.			
Conocimientos			
<ul style="list-style-type: none"> – Conocimientos en Administración de Sistemas Informáticos. – Conocimientos de los Gestores de Base de Datos PostgreSQL. – Conocimientos en Sistemas Webs basados en Java EE, específicamente JSF 2.X. – Manejo del Sistema Operativo basado en Linux de la rama de servidores. – Conocimientos de Administración General. – Conocimientos avanzados en el manejo del Sistema Operativo Linux, en cualquiera de sus distribuciones, Debian, Ubuntu, Linux Mint, etc. Sistema Operativo Windows en sus versiones XP, 7, 8 y 10. 			
Experiencia			
<ul style="list-style-type: none"> – Un año en labores de mantenimiento y administración general con énfasis en recursos humanos. – Un año en actividades de Administración y Desarrollo de Sistemas Informáticos. – Experiencia en el uso de sistemas web. 			
Cargo sugerido			
Jefe de Área de Soporte y Tecnología Informática.			

TABLA 4.1: Perfil del Administrador general del Sistema

Rol	Técnico del Área	Código	tecnA
Nivel Académico			
Técnico en Mecánica, Electricidad o Electrónica.			
Conocimientos			
<ul style="list-style-type: none"> – Conocimientos en electricidad básica y avanzada, mecánica básica, amplia variedad de equipos informáticos y su reparación y mantenimiento. – Conocimientos básicos de computadora. – Conocimientos en el manejo del Sistema Operativo Linux, en cualquiera de sus distribuciones, Debian, Ubuntu, Linux Mint, etc. Sistema Operativo Windows en sus versiones XP, 7, 8 y 10. 			
Experiencia			
Medio año en labores de mantenimiento.			
Cargo sugerido			
Técnico de Área de Soporte y Tecnología Informática.			

TABLA 4.2: Perfil del Técnico de Área

Rol	Solicitante	Código	solicit
Nivel Académico			
El nivel académico del solicitante puede variar según el nivel académico del personal de cada Unidad o Departamento en el cual tendrá acceso al Sistema Informático.			
Conocimientos			
<ul style="list-style-type: none"> – Conocimientos básicos en computadoras. – Conocimientos básicos en el Manejo de Sistemas Operativos Linux en sus diferentes Distribuciones: Debian, Ubuntu, Linux Mint, etc. Sistema Operativo Windows en sus versiones XP, 7, 8 y 10. 			
Experiencia			
No existe experiencia requerida para este rol.			
Cargo sugerido			
No existe cargo sugerido.			

TABLA 4.3: Perfil del Solicitante

4.1.2.4. Suposiciones y dependencias.

En esta subsección describirá aquellos factores que, si cambian, pueden afectar a los requisitos. Es una de las partes vitales, ya que el cambio o la falta de alguna de estas suposiciones generará un impacto en el proyecto puede afectar el cumplimiento exitoso de las actividades de este.

Acceso y ubicación del servidor del Sistema.

El lugar de alojamiento del servidor del Sistema será en un espacio en el servidor interno de la Facultad Multidisciplinaria de Occidente de la Universidad de El Salvador. El administrador general del Sistema y los solicitantes podrán acceder la desde cualquier navegador web de un equipo o dispositivo que esté conectado a la red interna de la Facultad mediante una dirección URL.

Solicitud del soporte técnico.

Cualquiera personal docente o administrativo con equipo asignado dentro de la Facultad podrá solicitar un servicio de soporte técnico mediante el uso del sistema

sin necesidad de iniciar ningún tipo de sesión o podrá utilizar comunicarse de otra forma y el personal del Área se encargará de crear la solicitud con los datos brindados.

4.1.3. Requisitos específicos.

La especificación de requerimientos de software es la descripción detallada de las necesidades y/o características que el Sistema ha de cubrir o soportar y lo que ha de realizar. De forma general, se define el comportamiento de este; así como, aquellos elementos con los que debe cumplir.

4.1.3.1. Interfaces externas.

Comprende el conjunto de conexiones físicas o funcionales que van desde interfaces de usuario, hasta protocolos de comunicación, entre otros.

Interfaces de usuario.

Su principal objetivo, consiste en proporcionar un entorno visual para permitir una fácil interacción entre el Sistema y el usuario. Los requisitos de las interfaces de usuario se presentan en la *TABLA 4.4*.

Código	Descripción
IUS-01	Interfaz de login para el administrador general del sistema.
IUS-02	El sistema estará optimizado para los navegadores web Iceweasel, Mozilla Firefox y Google Chrome.

TABLA 4.4: Interfaces de usuario

Interfaces de Software.

Especifican el uso de otros productos de software necesarios para el desarrollo, pruebas y puesta en marcha del Sistema. Las Interfaces de Software se presentan en el *TABLA 4.5*:

Código	Descripción
ISW-01	Para el desarrollo y administración de la base de datos, el sistema deberá utilizar el Gestor de Base de Datos PostgreSQL.
ISW-02	IDE Netbeans para programar y depurar el código del sistema.
ISW-03	Servidor de aplicaciones Glassfish 5.0 para comprobar la correcta ejecución y funcionamiento del sistema.
ISW-04	Git para el control de las modificaciones que se hagan durante todo el desarrollo en los archivos del sistema.

TABLA 4.5: Interfaces de Software

Interfaces de Comunicación.

Son las conexiones físicas y funcionales que permite la interacción entre diferentes dispositivos o sistemas. Su objetivo es definir los requerimientos de comunicación con relación a protocolos, puertos, etc. Las interfaces de comunicación se presentan en el *TABLA 4.6*:

Código	Descripción
ICO-01	Se requiere que la comunicación del sistema se realice por medio del protocolo TCP.
ICO-02	Se utilizará el puerto 8080 para el acceso web al sistema, por medio del protocolo HTTP.
ICO-03	El sistema será accesible solamente en la red interna de la Facultad Multidisciplinaria de Occidente de la Universidad de El Salvador, es decir, que no se podrá acceder a él por medio de Internet.
ICO-04	Para el control de versiones del software, en su desarrollo se utilizará Git.
ICO-05	Para la comunicación por medio de correo electrónico se utilizará el protocolo SMTP (Simple Mail Transfer Protocol).

TABLA 4.6: Interfaces de comunicación

Requerimientos de adaptación al sitio.

Los requerimientos de adaptación del sitio comprenden las necesidades de software, hardware y la configuración requeridas para que se cumpla una correcta funcionalidad del sitio. Los Requerimientos de Adaptación del Sitio se presentan en el *TABLA 4.7*:

Código	Descripción
RAS-01	Instalación y configuración correcta de Glassfish 5.0 como servidor HTTP (Servidor Web de Pruebas).

TABLA 4.7: Interfaces de adaptación al sitio

4.1.3.2. Funciones.

Definen las acciones específicas que debe proporcionar el Sistema y su comportamiento ante determinados escenarios, que van desde la captura de datos, el procesamiento de estos, y las respectivas salidas que debe entregar como resultado.

Requerimientos funcionales generales.

A continuación, en la *TABLA 4.8*, se describe de manera general los requerimientos funcionales que el Sistema debe cumplir, así como los actores que podrán realizar las acciones descritas.

Código	Descripción	Involucrados
RFU-01	El sistema deberá permitir el inicio y cierre de sesión.	adminGS
RFU-02	El sistema deberá permitir el cambio de contraseña.	adminGS
RFU-03	El sistema deberá permitir el cambio de datos de usuario.	adminGS
RFU-04	El sistema deberá permitir la administración de usuarios.	adminGS
RFU-05	El sistema deberá permitir la gestión de técnicos.	adminGS
RFU-06	El sistema deberá permitir la realización de solicitudes de servicio de mantenimiento.	adminGS, solicit
RFU-07	El sistema deberá permitir verificar el estado de la solicitud de servicio.	adminGS, solicit
RFU-08	El sistema deberá permitir la administración de equipos.	adminGS
RFU-09	El sistema deberá permitir el cambio del estado de la solicitud de soporte técnico.	adminGS
RFU-10	El sistema deberá permitir la consulta de las características del equipo.	adminGS
RFU-11	El sistema deberá permitir la consulta de la historial de mantenimiento del equipo.	adminGS
RFU-12	El sistema deberá permitir la generación del reporte de bitácora.	adminGS
RFU-13	El sistema deberá permitir la generación de listado de solicitudes de soporte técnico.	adminGS
RFU-14	El sistema deberá permitir la generación del informe de mantenimiento de una solicitud de soporte técnico.	adminGS
RFU-15	El sistema deberá permitir la generación de reportes de mantenimientos según los criterios de búsqueda especificados.	adminGS
RFU-16	El sistema permitirá almacenar la calendarización de los mantenimientos preventivos.	adminGS
RFU-17	El sistema permitirá la creación de cartas para notificar el mantenimiento preventivo a los diferentes Departamentos y Unidades.	adminGS
RFU-18	El sistema permitirá la creación de una memoria de labores.	adminGS
RFU-19	El sistema permitirá administrar una lista para las órdenes de compra.	adminGS

TABLA 4.8: Requerimientos funcionales generales

4.1.3.3. Atributos del sistema.

Se detallarán los atributos de calidad del sistema: fiabilidad, mantenibilidad, portabilidad, y, muy importante, la seguridad.

Atributo de calidad	Descripción
Confiabilidad	<ul style="list-style-type: none"> - El sistema tendrá una disponibilidad del 99% de las veces en que el usuario intente acceder. - El sistema será capaz de operar adecuadamente hasta con 100 usuarios conectados.
Facilidad de Uso	<ul style="list-style-type: none"> - La interfaz de usuario debe ser amigable. - La interfaz de usuario debe ser consistente en toda aplicación web y móvil. - El sistema mostrará mensajes de error que sean informativos. - El tiempo de capacitación por un usuario será menor a 1 hora.
Velocidad	<ul style="list-style-type: none"> - Toda funcionalidad del sistema responderá al usuario en menos de 5 segundos. - La información modificada en la base de datos debe ser actualizados en menos de 3 segundos. - El tiempo para iniciar o reiniciar el sistema será menor a 5 minutos.
Seguridad	<ul style="list-style-type: none"> - Los permisos y roles del sistema solamente podrán ser cambiados por el administrador. - El sistema permitirá la creación de respaldos.

TABLA 4.9: Requerimientos de calidad

4.1.3.4. Otros requisitos.

Requerimientos de Hardware.

Especifican las características técnicas necesarias del equipo cliente y servidor que permitirán la implementación del producto a entregar.

A continuación, en la *TABLA 4.10* se presentan las características y especificaciones técnicas mínimas del Hardware del Servidor donde estará alojado el servidor web y el sistema gestor de base de datos que contendrán tanto el código fuente y base de datos del sistema:

Característica	Especificación técnica
Modelo	-
Factor de Forma	-
Procesador	1.4 Ghz con arquitectura de 64 bits
Sockets de Procesadores	-
Front Side Bus/HyperTransport	-
Cache	-
Chipset	-
Memoria RAM	2 GB
I/O Slots	-
Disco Duro	500 GB
Conexiones de Red	Adaptador Ethernet (10/100/1000)
Disponibilidad	-
Fuente de Alimentación	-
Video	-
Sistema Operativo	Sistemas basados en Linux

TABLA 4.10: Requerimientos de Hardware

Requerimientos Mínimos de Máquina Cliente.

A continuación, en la *TABLA 4.11* se presentan los requerimientos mínimos necesarios de la máquina cliente, con los cuáles el sistema podrá funcionar adecuadamente:

Característica	Especificación técnica
Procesador	Intel® Pentium 4, 1.0GHz
Memoria RAM libre necesaria	128MB
Espacio Disco Duro libre necesario	1GB
Conexiones de Red	Tarjeta de Red Ethernet a 1 Gbit/s Tarjeta Inalámbrica 802.11b

TABLA 4.11: Requerimientos Mínimos de Máquina Cliente

4.2. Definición Tecnológica.

4.2.1. Comparativa entre tecnologías.

4.2.1.1. Lenguajes de programación.

Son lenguajes diseñados para que un conjunto de instrucciones pueda ser ejecutadas por la computadora, a cada una de estas instrucciones le corresponde una acción del procesador, llamado lenguaje máquina, todas estas instrucciones son ordenadas de forma lógica para solucionar un problema, luego se escriben o codifican estas soluciones por medio de un lenguaje de programación, al terminar se compilan todas las instrucciones, se realizan pruebas y depuramos el programa.

A continuación, se presenta en la *TABLA 4.12* una comparativa con los diferentes lenguajes de programación.

Lenguaje	Ventajas	Desventajas
JAVA	<ul style="list-style-type: none">- Es multiplataforma- Lenguaje Orientado a Objetos- Gestión de memoria automática- Facilidad de aprendizaje- Es gratuito- Documentación extensa	<ul style="list-style-type: none">- Rendimiento de ejecución menor debido a que es lenguaje interpretado- Necesita de una JVM (Java Virtual Machine) para ejecutar los programas- No es bueno como primera opción para principiantes en programación.- Sintaxis engorrosa en comparación con otros lenguajes.
PHP	<ul style="list-style-type: none">- Curva de aprendizaje muy corta- Soporta en cierta medida la orientación a objetos- Es un lenguaje multiplataforma- Documentación extensa	<ul style="list-style-type: none">- Necesita un servidor web para testear- Puede fomentar la creación de código desordenado- No tiene separación de capas- Dificulta la modularización.
Java Script	<ul style="list-style-type: none">- Es un lenguaje liviano- Recomendado para la creación de aplicaciones web- Utiliza poca memoria- Fácil manejo de datos- Soportado por casi todos los navegadores	<ul style="list-style-type: none">- Recursos no tan extensos, implementaciones de terceros requeridas.- Las implementaciones aumentan la curva de aprendizaje.- Los usuarios pueden desactivarlo en sus navegadores.
C++	<ul style="list-style-type: none">- Es multiplataforma- Curva de aprendizaje rápida- Es un lenguaje muy robusto- Documentación extensa	<ul style="list-style-type: none">- Necesita DLLs (Librerías Dinámicas) para funcionar correctamente.- No es recomendable para desarrollo web.- Gestión de memoria manual.- Creación de aplicaciones por capas engorrosa.

PYTHON	<ul style="list-style-type: none"> - Curva de aprendizaje rápida - Legibilidad del código para la lectura - Es código abierto - Facilidad de escritura de códigos asíncrono - Abundancia de bibliotecas - Documentación extensa 	<ul style="list-style-type: none"> - Es poco flexible - Rapidez de ejecución menor en comparación con otros lenguajes - Presenta problemas de seguridad
C#	<ul style="list-style-type: none"> - Estructurado - Orientado a objetos - Multiplataforma - Rango amplio y definido de tipos de datos - Control y ejecución simultánea de múltiples versiones - Flexibilidad 	<ul style="list-style-type: none"> - Requerimientos del sistema - Portabilidad - Carencia de un sistema de documentación universal - Entornos no utilizan el mismo framework para las pruebas.

TABLA 4.12: Comparativa entre lenguajes de programación

Java EE 7

El lenguaje de programación que se ha decidido usar para la codificación de la aplicación es Java, en su versión empresarial (Java EE) la cual permite el desarrollo de aplicaciones web, tomando como base los siguientes puntos:

- Por defecto, Java EE posee un stack completo de apis y librerías para desarrollar una aplicación multicapa completa, es decir que no es necesario utilizar frameworks de terceros en el frontend o backend como es el caso de utilizar PHP o JavaScript, lo que permite encapsular toda la aplicación bajo el mismo lenguaje nativo de Java y facilita la separación entre capas para mayor control y facilidad en el desarrollo.
- Para aplicaciones internas que se ejecutan en la intranet de las organizaciones Java EE es uno de los mejores candidatos para desarrollo por sus características de seguridad, estabilidad y escalabilidad.
- Actualmente en el mercado laboral es uno de los lenguajes de programación más cotizados y mejor pagados por las empresas tanto nacionales como extranjeras.
- Cuenta con mucha información, cursos, libros, videos y toda clase de documentación con lo cual la curva de aprendizaje se torna más rápida.

JPA

Java EE proporciona distintas soluciones para gestionar la capa de persistencia de las aplicaciones. La solución de más alto nivel es Java Persistence API, un framework que realiza un mapeo objeto-relacional y que permite trabajar esconder las tablas y las relaciones entre registros bajo una capa orientada a objetos en la que se definen entidades, campos y relaciones.

Se utilizará el API JPA para modelar y gestionar la persistencia de la aplicación. Se definirá el modelo de datos, mapeando las tablas de los modelos relacionales a clases y objetos. Además de utilizar el contexto de persistencia para trabajar con los datos en memoria. Por último, JPA proporciona la gestión la concurrencia en las peticiones.

EJB

Se utilizará EJB como una capa de abstracción que proporcionará una visión de objetos Java al modelo relacional existente en la base de datos.

En la capa de negocio se expone la lógica necesaria a la capa de presentación para que el usuario a través de la interfaz interactúe con las funcionalidades de la aplicación.

La plataforma Java EE define el uso de componentes de negocio EJB para abstraer la lógica de negocio de otros problemas generales de las aplicaciones empresariales como la concurrencia, transacciones, persistencia y seguridad.

JSF con PrimeFaces

PrimeFaces es una librería de componentes visuales open source que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las

aplicaciones web. Una de las ventajas de utilizar PrimeFaces, es que permite la integración con otros componentes.

La principal característica de PrimeFaces es la sencillez de instalación, es decir, sin dependencias ni configuraciones. Los componentes de PrimeFaces son amigables al usuario además de contar con un diseño innovador.

4.2.1.2. Gestores de base de datos.

Los sistemas de gestores de base de datos nos facilitan tanto la creación como el uso y mantenimiento de una base. Existen muchos gestores con distintas características que pueden ser de pago o gratuitos, por ejemplo, podemos mencionar Oracle Database, Microsoft SQL Server, PostgreSQL, MySQL, MariaDB, SQLite, MongoDB y CouchDB que son bases de datos no relacionales, Firebase que es un servicio que nos funciona como una base de datos remota alojada en la nube.

A continuación, se presenta una tabla comparativa sobre las diferentes alternativas disponibles en mercado sobre soluciones para el almacenamiento de datos.

Gestor	Ventajas	Desventajas
MySQL	<ul style="list-style-type: none"> - Instalación ilimitada y gratuita pues es de código abierto y multiplataforma. - Ofrece mayor velocidad al realizar consultas sencillas en bases de datos relativamente pequeñas. - Bajo costo en requerimientos pues puede ser ejecutado en máquinas con escasos recursos sin ningún problema. - Gran facilidad de uso y curva de aprendizaje rápida. 	<ul style="list-style-type: none"> - Cumplimiento parcial del estándar SQL. - Cumplimiento parcial de las características ACID. - Su propietario es Oracle. - Por defecto, no ofrece grandes opciones de escalabilidad.

PostgreSQL	<ul style="list-style-type: none"> - Instalación ilimitada y gratuita pues es de código abierto y multiplataforma. - Se puede configurar según el hardware de cada equipo. - Ofrece mayor velocidad para realizar análisis de datos en grandes volúmenes de datos. - Cumple con el estándar SQL. - Cumple con las características ACID. 	<ul style="list-style-type: none"> - Relativamente lento en inserciones y actualizaciones en base de datos pequeñas. - No cuenta con soporte en línea o telefónico. Todo se hace a través de foros oficiales. - La sintaxis de algunos comandos o sentencias no son intuitivas si no se tiene un nivel medio de conocimientos.
Oracle	<ul style="list-style-type: none"> - Motor de base de datos más usado a nivel mundial. - Posee herramientas para resolver casi cualquier problema empresarial. - Escalabilidad muy por encima de los demás motores de bases de datos. - Cumple por completo el estándar SQL. - Cumple las características ACID. 	<ul style="list-style-type: none"> - El elevado costo económico en licencias personales. - Una inadecuada configuración en la instalación de este software lo convierte en un sistema desesperante y lento. - Curva de aprendizaje muy lenta.
Microsoft SQL Server	<ul style="list-style-type: none"> - Características de compresión que reducen los requisitos necesarios para el almacenamiento de los datos. - Gran escalabilidad empresarial. - Facilidad de instalación. - Facilidad de uso y curva de aprendizaje moderada. - Cumple por completo el estándar SQL. - Cumple con las características ACID. 	<ul style="list-style-type: none"> - Por defecto, consumo masivo de memoria RAM para su uso. - No es multiplataforma. - Relación costo-rendimiento muy por debajo de Oracle.

TABLA 4.13: Comparativa entre gestores de base de datos

PostgreSQL

Para proporcionar la gestión de bases de datos para el manejo de la información del Sistema Informático se opta por utilizar PostgreSQL ya que sus características técnicas lo hacen uno de los gestores más potentes y robustos del mercado. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al **sistema**. Además, se ha tomado en cuenta los siguientes puntos:

- Es un sistema multiplataforma que posee extensiones para trabajar con Java, además gracias a su gran escalabilidad nos permite ajustarse al número de CPU y a la cantidad de memoria disponible de forma óptima.
- Tiene soporte para JSON algo que otros gestores de BD relacionales no manejan bien, se puede realizar copias de seguridad en caliente, además de poseer con una completa documentación lo cual facilita y disminuye el tiempo de aprendizaje para utilizarlo.
- Alta concurrencia. PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit.
- Permite estabilidad y confiabilidad en contraste a muchos sistemas de bases de datos comerciales.
- Es extensible ya que el código fuente está disponible sin costo. Si se necesita extender o personalizar PostgreSQL, puede hacerse con un mínimo esfuerzo y sin costos adicionales.
- Facilidad de manejo. PgAdmin es el principal administrador de base de datos de Postgres, es muy sencillo de manejar, siendo óptima para la realización del proyecto. Además, PgAdmin brinda la facilidad para hacer mantenimiento de las tablas o respaldos.

4.2.2. Tecnologías complementarias.

4.2.2.1 Netbeans IDE.

NetBeans IDE es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender sus funcionalidades. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

Para el desarrollo de este sistema se utilizará Netbeans IDE 8.2

4.2.2.2 Glassfish.

GlassFish es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, compañía adquirida por Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación.

Para el desarrollo de este sistema se utilizará Glassfish 5.0

4.2.2.3 Git.

Es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

4.3 Diseño del software.

Se presentará el diseño de los diferentes elementos del sistema para el Área de Soporte y Tecnología Informática. Se definirán las técnicas y modelos de análisis y diseño para la programación, en relación con el enfoque a utilizar para el desarrollo. El diseño de la base de datos es la estructura de los datos que tendrá el sistema, siguiendo las fases de su proceso de diseño, definiendo para ello el modelo conceptual y lógico. Además de los elementos antes mencionados, también involucra el diseño de interacción del sistema, estableciendo el comportamiento que éste tendrá ante la interacción con los usuarios.

4.3.1. Técnicas de programación.

En relación con el modelo a utilizar en el desarrollo del software y las técnicas de análisis y diseño seleccionadas, se utiliza la Programación Orientada a Objetos para la etapa de codificación.

Por medio de las técnicas de programación se escoge de forma lógica como se determinará la ejecución de un software.

4.3.1.1. Programación Orientada a Objetos.

La programación orientada a objetos (POO) es un paradigma de programación que usa los objetos en sus interacciones. Está basado en varias características, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. (Pérez, María. 2012, Programación Orientada a Objetos y Programación Estructurada).

Algunas definiciones básicas de esta técnica de programación son las siguientes:

- Clase: Define las propiedades y comportamientos generales de un tipo de objeto concreto. La instanciación es una copia de estas propiedades y comportamientos, y la creación de un objeto a partir de ellas.
- Objeto: Instancia de una clase. Entidad provista de un conjunto de propiedades o atributos y de comportamiento o funcionalidad, los mismos que consecuentemente reaccionan a eventos.
- Método: Algoritmo o función asociada a un objeto o a una clase de objetos, cuya ejecución se desencadena tras la recepción de un "mensaje".
- Evento: es un suceso en el sistema, una interacción del usuario con la máquina, o un mensaje enviado por un objeto.
- Atributos: Propiedades o características que tiene la clase.
- Mensaje: Una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.
- Herencia: Es la relación entre una clase general y otra clase más específica, implica que una subclase obtiene todos los métodos y eventualmente las variables de su superclase.
- Abstracción: Características esenciales de un objeto, donde se capturan sus comportamientos.

- Encapsulamiento: Significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción.

(Durán, Francisco. Gutiérrez, Francisco. Pimentel, Ernesto. 2007. Programación orientada a objetos con Java)

4.3.2. Diseño Arquitectónico.

El Diseño Arquitectónico da un punto de vista del sistema que permite verlo como un todo, y a la vez constituye el punto de partida de los subsiguientes modelos de diseño (como el modelo de clases o el modelo de datos), siendo esta una de las herramientas de comunicación entre los interesados en el desarrollo del sistema (Pressman, R. S. (2010). Ingeniería del software, Un Enfoque Práctico. 7ª Edición. McGraw-Hill). Por tanto, el diseño de la arquitectura del sistema tendrá un efecto profundo en el desarrollo de todo el proyecto, porque los resultados de esta, guiarán todos los demás aspectos del diseño.

4.3.2.1. Modelo de Tres Capas.

Es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica del negocio, los datos y la presentación al usuario (Interfaz de Usuario).

- Capa de presentación: Es la capa con la cual el usuario interactúa, por lo cual también se le denomina capa de usuario, ya que presenta al usuario la información solicitada y captura los datos ingresados por él. Esta capa se comunica únicamente con la capa de negocio.
- Capa de negocio: Se denomina también capa de lógica del negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes o peticiones de usuario y presentar los resultados o respuestas a dichas solicitudes, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

- Capa de datos: Es donde residen los datos, recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

En la *FIGURA 4.1* se representa gráficamente el Modelo de Tres Capas y la relación entre estas. El sistema funcionará en un ambiente web, por lo cual los usuarios podrán acceder al sistema mediante un navegador web desde un equipo cliente localizado en cualquier establecimiento de la Facultad, el cual hará el papel de capa de presentación, desde aquí se realizarán solicitudes o peticiones a un servidor web el cual hace el papel de capa de negocio, este a su vez se comunica con el sistema gestor de base de datos para poder satisfacer las peticiones solicitadas desde la capa de presentación.

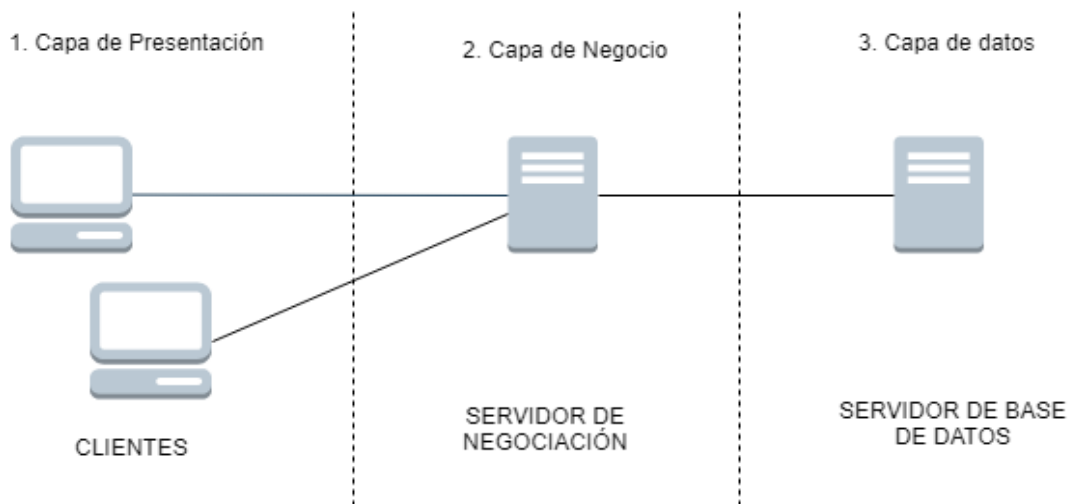


FIGURA 4.1: Modelo de Tres Capas (Fuente Propia)

4.3.2.2. Patrón Modelo Vista Controlador.

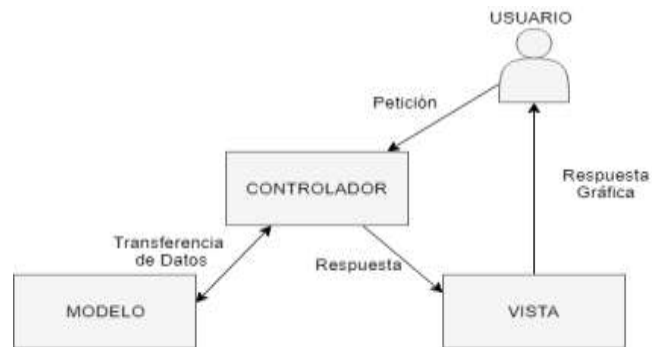
El Patrón o Arquitectura Modelo Vista Controlador (MVC), separa la presentación y la interacción de los datos del sistema. El sistema se estructura en tres componentes lógicos que interactúan entre sí.

El componente modelo gestiona los datos del sistema y las operaciones asociadas a esos datos. La componente vista define y administra la forma en que se presentan los datos para el usuario. El componente controlador gestiona la interacción del usuario, es decir, sus peticiones o solicitudes (por medio de selección de opciones de menú, clics sobre botones,

pulsación de teclas, etc.) y pasa estas interacciones al modelo en caso de ser necesario, y finalmente retornando una respuesta a la vista, además de realizar el procesamiento y otros cálculos.

A simple vista, el patrón MVC podría confundirse con el modelo anterior, sin embargo, el patrón MVC, sostiene separar los componentes de un sistema (separar los elementos visuales/gráficos de su correspondiente programación y acceso a datos), mientras que el modelo de tres capas organiza los componentes de un sistema con relación a su implementación física.

En la *FIGURA 4.2* se representa gráficamente el patrón MVC:



**FIGURA 4.2: Modelo Vista Controlador
(Fuente Propia)**

4.4. Diseño de la base de datos.

4.4.1. Diagrama Entidad – Relación.

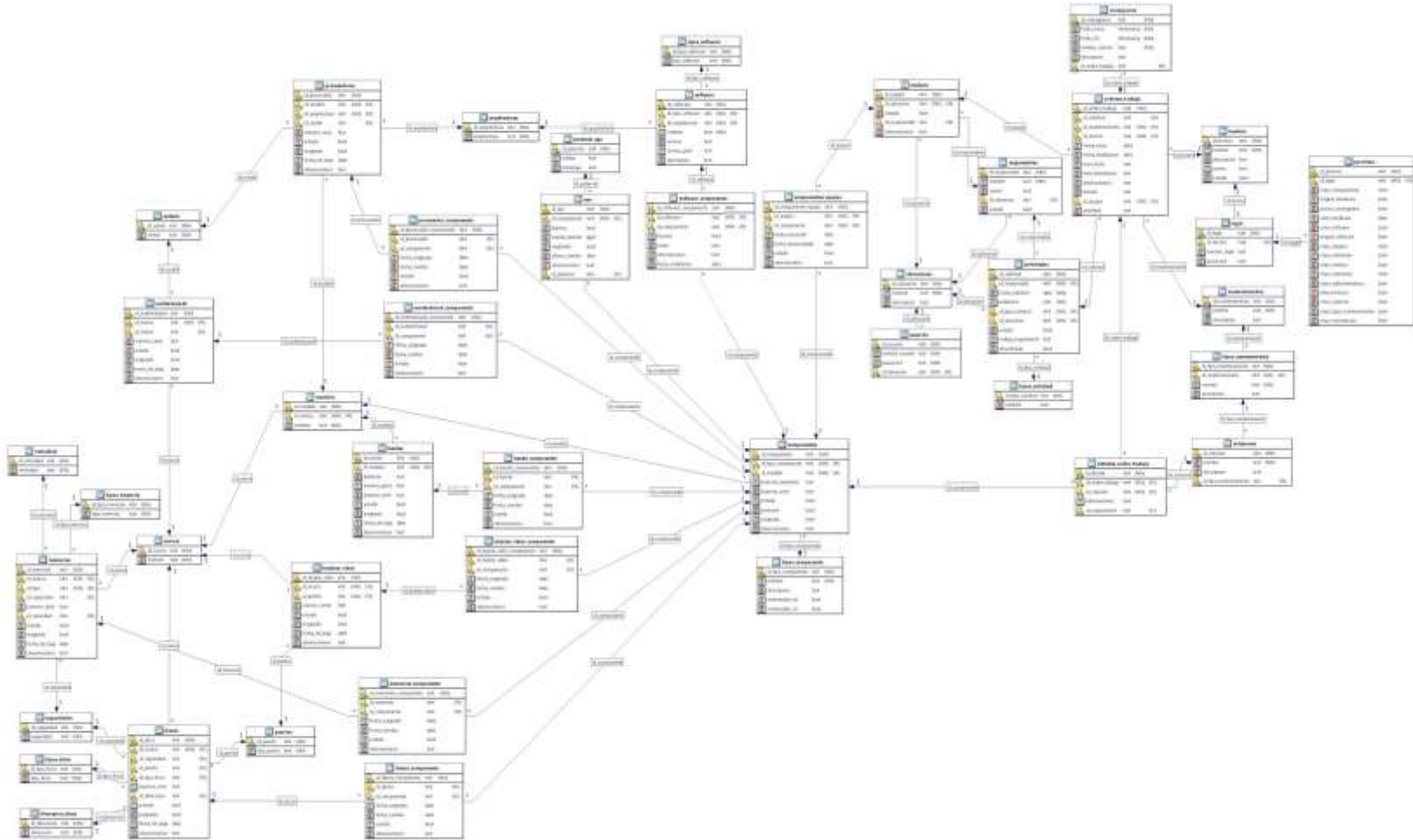


FIGURA 4.3: Diagrama ER

4.4.2. Diccionario de Datos.

A continuación, se presentan todas las definiciones para las tablas y tipos de datos que se utilizarán en el sistema:

Nombre: arquitecturas			
Descripción: Almacenará la información de las diferentes arquitecturas de algunos componentes y de los softwares, por ejemplo, si son de 32 o 64 bits			
Llave	Campo	Tipo	Descripción
PK	id_arquitectura	serial	Auto-incrementable que servirá como llave primaria
	arquitectura	text	Nombre distintivo de la arquitectura
Relaciones: --			

Nombre: capacidades			
Descripción: Almacenará la información de las diferentes capacidades que poseen los componentes, por ejemplo, 500GB, 1TB			
Llave	Campo	Tipo	Descripción
PK	id_capacidad	serial	Auto-incrementable que servirá como llave primaria
	capacidad	text	Capacidad del componente
Relaciones: --			

Nombre: dimension_disco			
Descripción: Almacenará las diferentes dimensiones que los discos duros pueden poseer			
Llave	Campo	Tipo	Descripción
PK	id_dimension	serial	Auto-incrementable que servirá como llave primaria
	dimension	text	Almacenará la dimensión del disco
Relaciones: --			

Nombre: componentes			
Descripción: Almacenará la información específica de cada componente que se tiene dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_componente	serial	Auto-incrementable que servirá como llave primaria
FK	id_tipo_componente	integer	Identificador del tipo al que pertenece el componente
FK	id_modelo	integer	Identificador del modelo del componente
	numero_inventario	text	Número que se utiliza para manejo interno del inventario dentro de la Facultad
	numero_serie	text	Serial único del fabricante
	estado	boolean	Determina si el componente se encuentra en funcionamiento o ya se dio de baja
	personal	boolean	Determina si el componente es de uso personal
	asignado	boolean	Determina si el componente se encuentra asignado
	observaciones	text	Información adicional que se desea almacenar
Relaciones: <i>id_modelo</i> con <i>modelos</i> <i>id_tipo_componente</i> con <i>tipos_componente</i>			

Nombre: potencia_ups			
Descripción: Almacenará la información de las potencias que pueden tener los ups dentro de la Facultad.			
Llave	Campo	Tipo	Descripción
PK	id_potencia	serial	Auto-incrementable que servirá como llave primaria
	voltaje	text	Almacenará el voltaje
	amperaje	text	Almacenará el amperaje
Relaciones: --			

Nombre: componentes_equipo			
Descripción: Almacenará la información sobre el software que tiene instalado cada CPU dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_componente_equipo	serial	Auto-incrementable que servirá como llave primaria
FK	id_equipo	integer	Identificador del equipo
FK	id_componente	integer	Identificador del componente
	fecha_vinculado	date	Especifica la fecha en que se asignó el componente al equipo
	fecha_desvinculado	date	Especifica la fecha en que se desvinculó el componente al equipo
	estado	boolean	Indica que está activo o inactivo
	observaciones	text	Información adicional que se desea almacenar
Relaciones: <i>id_equipo con equipos</i> <i>id_componente con componentes</i>			

Nombre: modelos			
Descripción: Almacenará la información de los diferentes modelos de componentes que se manejan dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_modelo	serial	Auto-incrementable que servirá como llave primaria
FK	id_marca	integer	Identificador de la marca a la que pertenece el modelo
	modelo	text	Nombre o número distintivo del modelo
Relaciones: <i>id_marca con marcas</i>			

Nombre: cronograma			
Descripción: Almacenará las diferentes actividades programadas			
Llave	Campo	Tipo	Descripción
PK	id_cronograma	serial	Auto-incrementable que servirá como llave primaria
FK	id_orden_trabajo	integer	Asocia con la orden de trabajo
	fecha_inicio	date	Almacena la fecha de inicio de la actividad
	fecha_fin	date	Almacena la fecha fin de la actividad
	nombre_evento	text	Almacena el nombre de la actividad
	descripcion	text	Información acerca de la actividad
Relaciones: <i>id_orden_trabajo con ordenes_trabajo</i>			

Nombre: detalles_orden_trabajo			
Descripción: Almacenará los detalles del trabajo realizado durante el mantenimiento al equipo.			
Llave	Campo	Tipo	Descripción
PK	id_detalle	serial	Auto-incrementable que servirá como llave primaria
FK	id_orden_trabajo	integer	Asocia a ordenes_trabajo, establece la solicitud que se está trabajando
FK	id_componente	integer	Asocia a componente de orden de trabajo
FK	id_solucion	integer	Asocia con soluciones, indica la solución que se dio ante el mantenimiento.
	observaciones	text	Información adicional acerca de la orden de trabajo
Relaciones: <i>id_orden_trabajo con ordenes_trabajo</i> <i>id_componente con componentes</i> <i>id_solucion con soluciones</i>			

Nombre: discos			
Descripción: Almacenará las características de los diferentes discos duros que se manejan dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_disco	serial	Auto-incrementable que servirá como llave primaria
FK	id_marca	integer	Identificador de la marca a la que pertenece el disco
FK	id_tipo_disco	integer	Identificador del tipo al que pertenece el disco
FK	id_capacidad	integer	Identificador de la capacidad que tiene el disco
FK	id_puerto	integer	Identificador del puerto de conexión que tiene el disco
	numero_serie	text	Serial del fabricante del disco
FK	id_dimension	integer	Identificador de la dimensión que tiene el disco
	estado	boolean	Indica que está activo o inactivo
	asignado	boolean	Identifica si el disco está asignado o no
	fecha_de_baja	date	Almacena la fecha en la que se le dio de baja al disco
	observaciones	text	Información adicional que se desea almacenar
Relaciones: <i>id_marca</i> con <i>marcas</i> <i>id_capacidad</i> con <i>capacidades</i> <i>id_puerto</i> con <i>puertos</i> <i>id_tipo_disco</i> con <i>tipos_disco</i> <i>id_dimension</i> con <i>dimension_disco</i>			

Nombre: discos_componentes			
Descripción: Almacenará la información de los discos que pertenecen o han pertenecido a un componente.			
Llave	Campo	Tipo	Descripción
PK	id_discos_compone nte	serial	Auto-incrementable que servirá como llave primaria
FK	id_discos	integer	Identificador del disco asociado al componente
FK	id_componente	integer	Identificador del componente que al cual se asociaron los discos
	fecha_asignado	date	Almacena la fecha en la que se asignó el disco al componente
	fecha_cambio	date	Almacena la fecha en la que se sustituyó el disco al componente
	estado	boolean	Indica si está activo o inactivo
	observaciones	text	Información adicional que se desea almacenar
Relaciones: <i>id_disco con discos</i> <i>id_componente con componentes</i>			

Nombre: mantenimientos			
Descripción: Almacenará los tipos de mantenimiento que da el área a la Facultad.			
Llave	Campo	Tipo	Descripción
PK	id_mantenimiento	serial	Auto-incrementable que servirá como llave primaria
	nombre	text	Nombre del mantenimiento
	descripcion	text	Información acerca de los mantenimientos
Relaciones: --			

Nombre: equipos			
Descripción: Almacenará la información de los equipos informáticos que se encuentran dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_equipo	serial	Auto-incrementable que servirá como llave primaria
FK	id_ubicacion	integer	Identificador para saber en qué ubicación se encuentra el equipo
FK	id_responsable	integer	Identificador para saber el responsable del equipo
	estado	boolean	Indica que está activo o inactivo
	observaciones	text	Información adicional sobre los equipos
Relaciones: <i>id_ubicacion con ubicaciones</i> <i>id_responsable con responsables</i>			

Nombre: login			
Descripción: Almacenará el usuario para que pueda iniciar sesión en el sistema			
Llave	Campo	Tipo	Descripción
PK	id_login	serial	Auto-incrementable que servirá como llave primaria
FK	id_tecnico	integer	Identificador del técnico que iniciara sesión
	nombre_login	text	Almacena el usuario de login
	password	text	Almacena la contraseña del usuario
Relaciones: <i>id_tecnico con tecnico</i>			

Nombre: fuente_componente			
Descripción: Almacenará la información de las fuentes que pertenecen o han pertenecido a un componente.			
Llave	Campo	Tipo	Descripción
PK	id_fuente_componente	serial	Auto-incrementable que servirá como llave primaria
FK	id_fuente	integer	Identificador de l fuente que se asociara
FK	id_componente	integer	Identificador del componente al que pertenece la fuente
	fecha_asignado	date	Fecha en que fue asignada dicha fuente al componente
	fecha_cambio	date	Fecha en que se cambió dicha fuente
	estado	boolean	Indica que está activo o inactivo
	observaciones	text	Información adicional sobre la asignación de la fuente
Relaciones: <i>id_fuente con fuentes</i> <i>id_componente con componentes</i>			

Nombre: marcas			
Descripción: Almacenará la información de las diferentes marcas que se manejan dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_marca	serial	Auto-incrementable que servirá como llave primaria
	marca	text	Nombre distintivo del tipo de la marca
Relaciones: --			

Nombre: fuentes			
Descripción: Almacenará las características de las diferentes fuentes de poder que se manejan dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_fuente	serial	Auto-incrementable que servirá como llave primaria
FK	id_modelo	integer	Identificador del modelo al que pertenece la fuente
	potencia	text	Especifica la potencia de watts que tiene la fuente
	numero_pines	text	Especifica el número de pines del conector de la fuente
	numero_serie	text	Serial del fabricante de la fuente
	estado	boolean	Indica si está activo o inactivo
	asignado	boolean	Indica si está asignado o no
	fecha_de_baja	date	Almacena la fecha en la que se dio de baja la fuente del componente
	observaciones	text	Información adicional que se desea almacenar
Relaciones: <i>id_modelo con modelos</i>			

Nombre: puertos			
Descripción: Almacenará la información de los diferentes tipos de puertos que poseen los componentes, por ejemplo, SATA, PCI, PCI-e			
Llave	Campo	Tipo	Descripción
PK	id_puerto	serial	Auto-incrementable que servirá como llave primaria
	tipo_puerto	text	Nombre distintivo del tipo de puerto
Relaciones: --			

Nombre: memorias			
Descripción: Almacenará las características de las diferentes memorias RAM que se manejan dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_memoria	serial	Auto-incrementable que servirá como llave primaria
FK	id_marca	integer	Identificador de la marca a la que pertenece la memoria
FK	id_tipo	integer	Identificador del tipo al que pertenece la memoria
FK	id_capacidad	integer	Identificador de la capacidad que tiene la memoria
FK	id_velocidad	integer	Identificador de la velocidad que tiene la memoria
	numero_serie	text	Serial del fabricante de la memoria
	estado	boolean	Indica si está activo o inactivo
	asignado	boolean	Indica si está asignado o no
	fecha_de_baja	date	Almacena la fecha en la que se dio de baja la memoria
	observaciones	text	Información adicional que se desea almacenar
Relaciones: <i>id_marca con marcas</i> <i>id_tipo con tipos_memoria</i> <i>id_capacidad con capacidades</i> <i>id_velocidad con velocidad</i>			

Nombre: tipos_software			
Descripción: Almacenará la información sobre los diferentes tipos de software que se utilizan dentro de la Facultad, por ejemplo, S.O, utilitarios, etc.			
Llave	Campo	Tipo	Descripción
PK	id_tipo_software	serial	Auto-incrementable que servirá como llave primaria
	tipo_software	text	Nombre con el que se identificará el tipo de software
Relaciones: --			

Nombre: memorias_componente

Descripción: Almacenará la información de las memorias que pertenecen o han pertenecido a un componente.

Llave	Campo	Tipo	Descripción
PK	id_memoria	serial	Auto-incrementable que servirá como llave primaria
FK	id_marca	integer	Identificador de la marca a la que pertenece la memoria
FK	id_tipo	integer	Identificador del tipo al que pertenece la memoria
FK	id_capacidad	integer	Identificador de la capacidad que tiene la memoria
FK	id_velocidad	integer	Identificador de la velocidad que tiene la memoria
	numero_serie	text	Serial del fabricante de la memoria
	estado	boolean	Indica si está activo o inactivo
	asignado	boolean	Indica si está asignado o no
	fecha_de_baja	date	Almacena la fecha en la que se dio de baja la memoria
	observaciones	text	Información adicional que se desea almacenar

Relaciones:

id_marca con *marcas*

id_tipo con *tipos_memoria*

id_capacidad con *capacidades*

id_velocidad con *velocidad*

Nombre: motherboard_componente			
Descripción: Almacenará la información de las tarjetas madre que pertenecen o han pertenecido a un componente.			
Llave	Campo	Tipo	Descripción
PK	id_motherboard_componente	serial	Auto-incrementable que servirá como llave primaria
FK	id_motherboard	integer	Identificador de la tarjeta madre que pertenece al componente
FK	id_componente	integer	Identificador del componente
	fecha_asignado	text	Fecha en que fue asignada dicha tarjeta madre al componente
	fecha_cambio	date	Fecha en que se cambió dicha tarjeta madre
	estado	boolean	Indica que está activo o inactivo
	observaciones	text	Información adicional sobre la asignación de la fuente
Relaciones: <i>id_motherboard</i> con <i>motherboards</i> <i>id_componente</i> con <i>componentes</i>			

Nombre: tipos_memoria			
Descripción: Almacenará la información de los diferentes tipos de memorias RAM que se utilizan dentro de la Facultad, por ejemplo, si son DDR1, DDR2 o DDR3			
Llave	Campo	Tipo	Descripción
PK	id_tipo_memoria	serial	Auto-incrementable que servirá como llave primaria
	tipo_memoria	text	Nombre distintivo del tipo de memoria
Relaciones: --			

Nombre: motherboards			
Descripción: Almacenará las características de las diferentes placas madre que se manejan dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_motherboard	serial	Auto-incrementable que servirá como llave primaria
FK	id_marca	integer	Identificador de la marca a la que pertenece la placa
FK	id_socket	integer	Identificador del socket de conexión que tiene la placa
	numero_serie	text	Serial del fabricante de la placa
	estado	boolean	Indica si está activo o inactivo
	asignado	boolean	Indica si está asignado o no
	fecha_de_baja	date	Almacena la fecha en la que se dio de baja la placa
	observaciones	text	Información adicional que se desea almacenar
Relaciones: <i>id_marca con marcas</i> <i>id_socket con sockets</i>			

Nombre: tipos_solicitud			
Descripción: Almacenará la información sobre los diferentes tipos de solicitud que se realizan dentro de la Facultad.			
Llave	Campo	Tipo	Descripción
PK	id_tipo_solicitud	serial	Auto-incrementable que servirá como llave primaria
	nombre	text	Nombre con el que se identificará el tipo de solicitud
Relaciones: --			

Nombre: ordenes_trabajo			
Descripción: Almacenará los mantenimientos en los que los técnicos están trabajando			
Llave	Campo	Tipo	Descripción
PK	id_orden_trabajo	serial	Auto-incrementable que servirá como llave primaria
FK	id_solicitud	integer	Asocia con solicitud, establece la solicitud que se trabaja
FK	id_mantenimiento	integer	Asocia con mantenimiento, permite saber el tipo de mantenimiento que se realizará
FK	id_tecnico	integer	Asocia con técnicos, establece la persona a cargo de la orden de trabajo
	prioridad	integer	Orden de prioridad que tendrá dentro de los mantenimientos
	fecha_inicio	dateTime	Fecha de inicio de orden de trabajo
	fecha_finalizacion	dateTime	Fecha de finalización de orden de trabajo
	hora_inicio	dateTime	Hora de inicio de orden de trabajo
	hora_finalizacion	dateTime	Hora de finalización de orden de trabajo
FK	id_equipo	integer	Asocia con el equipo de la orden de trabajo.
	estado	boolean	Indica si está activo o inactivo
	observaciones	text	Información adicional sobre la orden de trabajo realizada
Relaciones: <i>id_solicitud</i> con <i>solicitudes</i> <i>id_mantenimiento</i> con <i>mantenimientos</i> <i>id_tecnico</i> con <i>tecnicos</i> <i>id_equipo</i> con <i>equipos</i>			

Nombre: permisos			
Descripción: Almacenará los accesos de los usuarios en el sistema.			
Llave	Campo	Tipo	Descripción
PK	id_permiso	serial	Auto-incrementable que servirá como llave primaria
FK	id_login	integer	Identificador de la sesión activa
	crear_componentes	boolean	Indica si tiene acceso a la creación de componentes
	asignar_hardware	boolean	Indica si tiene acceso a la asignación de hardware
	acceso_cronograma	boolean	Indica si tiene acceso para ver el cronograma
	crear_hardware	boolean	Indica si tiene acceso a la creación de hardware
	asignar_software	boolean	Indica si tiene acceso a la asignación de software
	crear Equipos	boolean	Indica si tiene acceso a la creación de equipos
	crear_solicitudes	boolean	Indica si tiene acceso a la creación de solicitudes
	crear_ordenes	boolean	Indica si tiene acceso a la creación de órdenes de trabajo
	crear_soluciones	boolean	Indica si tiene acceso a la creación de soluciones
	crear_administradores	boolean	Indica si tiene acceso a la creación de usuarios administradores
	crear_tecnicos	boolean	Indica si tiene acceso a la creación de técnicos
	crear_externos	boolean	Indica si tiene acceso a la creación de usuarios externos
	crear_tipos_mantenimiento	boolean	Indica si tiene acceso a la creación de tipos de mantenimiento
	crear_miscelaneas	boolean	Indica si tiene acceso a la creación de misceláneas
Relaciones: <i>id_login con login</i>			

Nombre: procesador_componente			
Descripción: Almacenará la información de los procesadores que pertenecen o han pertenecido a un componente.			
Llave	Campo	Tipo	Descripción
PK	id_procesador_componente	serial	Auto-incrementable que servirá como llave primaria
FK	id_procesador	integer	Identificador procesador que pertenece al componente
FK	id_componente	integer	Identificador del componente
	fecha_asignado	text	Fecha en que fue asignado dicho procesador al componente
	fecha_cambio	date	Fecha en que se cambió dicho procesador
	estado	boolean	Indica que está activo o inactivo
	observaciones	text	Información adicional sobre la asignación de la fuente
Relaciones: <i>id_procesador</i> con <i>procesadores</i> <i>id_componente</i> con <i>componentes</i>			

Nombre: responsables			
Descripción: Almacenará la información básica de todas las personas que disponen de un equipo informático dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_responsable	serial	Auto-incrementable que servirá como llave primaria
	nombre	text	Nombre completo del responsable
	correo	text	Correo para contactar con el responsable
FK	id_ubicacion	integer	Identificador de la ubicación a la que pertenece el responsable
	estado	boolean	Indica si está activo o inactivo
Relaciones: <i>id_ubicacion</i> con <i>ubicaciones</i>			

Nombre: procesadores			
Descripción: Almacenará las características de los diferentes procesadores que se manejan dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_procesador	serial	Auto-incrementable que servirá como llave primaria
FK	id_modelo	integer	Identificador del modelo al que pertenece el procesador
FK	id_arquitectura	integer	Identificador de la arquitectura que tiene el procesador
FK	id_socket	integer	Identificador del socket de conexión del procesador
	numero_serie	text	Serial del fabricante del procesador
	estado	boolean	Indica si está activo o inactivo
	asignado	boolean	Indica si está asignado o no
	fecha_de_baja	date	Almacena la fecha en la que se dio de baja el procesador
	observaciones	text	Información adicional que se desea almacenar
Relaciones: <i>id_modelo</i> con <i>modelos</i> <i>id_arquitectura</i> con <i>arquitecturas</i> <i>id_socket</i> con <i>sockets</i>			

Nombre: sockets			
Descripción: Almacenará la información de los diferentes tipos de sockets que poseen los componentes, por ejemplo, AM2, AM3, LGA			
Llave	Campo	Tipo	Descripción
PK	id_socket	serial	Auto-incrementable que servirá como llave primaria
	socket	text	Nombre distintivo del tipo de socket
Relaciones: --			

Nombre: software			
Descripción: Almacenará la información específica de cada software que se manejan dentro de la Facultad con sus respectivas características			
Llave	Campo	Tipo	Descripción
PK	id_software	serial	Auto-incrementable que servirá como llave primaria
FK	id_tipo_software	integer	Identificador del tipo de software que es
FK	id_arquitectura	integer	Identificador para la arquitectura del software
	nombre	text	Nombre distintivo del software
	version	text	Versión a la que corresponde el software
	service_pack	text	Versión del service pack que utiliza
	descripcion	text	Información adicional que se desea almacenar
Relaciones: <i>id_tipo_software</i> con <i>tipos_software</i> <i>id_arquitectura</i> con <i>arquitecturas</i>			

Nombre: tipos_mantenimiento			
Descripción: Almacenará la clasificación de mantenimientos que prestan en el área de soporte y mantenimiento			
Llave	Campo	Tipo	Descripción
PK	id_tipo_mantenimient o	serial	Auto-incrementable que servirá como llave primaria
FK	id_mantenimiento	integer	Identificador del manteamiento al cual se asociará
	nombre	text	Nombre con el que se identificará el tipo
	descripción	text	Información adicional que se desea almacenar
Relaciones: <i>id_mantenimiento</i> con <i>mantenimientos</i>			

Nombre: softwares_componentes			
Descripción: Almacenará la información sobre el software que tiene instalado cada CPU dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_software_componente	serial	Auto-incrementable que servirá como llave primaria
FK	id_software	integer	Identificador del software
FK	id_componente	integer	Identificador del componente
	licencia	boolean	Especifica si el software instalado es original
	serial	text	Almacena el serial o key del software
	observaciones	text	Información adicional que se desea almacenar
	fecha_instalacion	date	Almacena la fecha en la que se instaló el software
Relaciones: <i>id_software</i> con <i>software</i> <i>id_componente</i> con <i>componentes</i>			

Nombre: soluciones			
Descripción: Almacenará un listado de las reparaciones que se pueden dar al equipo.			
Llave	Campo	Tipo	Descripción
PK	id_solucion	serial	Auto-incrementable que servirá como llave primaria
FK	id_tipo_mantenimiento	integer	Identificador del tipo de mantenimiento que se le realizará
	nombre	text	Nombre descriptivo de solución o reparación
	descripcion	text	Información acerca del tipo de reparación que se realizó
Relaciones: <i>id_tipo_mantenimiento</i> con <i>tipos_mantenimiento</i>			

Nombre: solicitudes			
Descripción: Almacenará la información de las solicitudes de mantenimientos y soporte técnico que se reciben dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_solicitud	serial	Auto-incrementable que servirá como llave primaria
FK	id_responsable		Identificador de la persona a la que pertenece el equipo
FK	id_tipo_solicitud	integer	Identificador del equipo al que se le realizará mantenimiento
FK	id_ubicacion	integer	Identificador del estado en que se encuentra el proceso de la solicitud
	estado	boolean	Indica si ya tiene una orden de trabajo asignada
	fecha_solicitud	date	Fecha en que se realiza la solicitud
	problema	text	Explicación del problema que presenta el equipo
	codigo_seguimiento	text	Almacena el código de la solicitud
	desactivada	boolean	Indica si solicitud se inactivara
Relaciones: <i>id_responsable</i> con <i>responsables</i> <i>id_tipo_solicitud</i> con <i>tipos_solicitud</i> <i>id_ubicacion</i> con <i>ubicaciones</i>			

Nombre: tipos_disco			
Descripción: Almacenará la información de los diferentes tipos de discos duros que se utilizan dentro de la Facultad, por ejemplo, HDD o SSD			
Llave	Campo	Tipo	Descripción
PK	id_tipo_disco	serial	Auto-incrementable que servirá como llave primaria
	tipo_disco	text	Nombre distintivo del tipo de disco
Relaciones: --			

Nombre: tarjetas_video			
Descripción: Almacenará las características de las diferentes tarjetas de video que se manejan dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_tarjeta_video	serial	Auto-incrementable que servirá como llave primaria
FK	id_marca	integer	Identificador de la marca a la que pertenece la tarjeta
FK	id_puerto	integer	Identificador del puerto de conexión al que pertenece la tarjeta
	numero_serie	text	Serial del fabricante de la tarjeta
	estado	boolean	Indica si está activo o inactivo
	asignado	boolean	Indica si está asignado o no
	fecha_de_baja	date	Almacena la fecha en la que se dio de baja la tarjeta de video
	observaciones	text	Información adicional que se desea almacenar
Relaciones: <i>id_marca</i> con <i>marcas</i> <i>id_puerto</i> con <i>puertos</i>			

Nombre: tipos_componente			
Descripción: Almacenará la información sobre los tipos de componentes externos que conforman un equipo informático como lo son monitores, CPU, teclados, etc.			
Llave	Campo	Tipo	Descripción
PK	id_tipo_componente	serial	Auto-incrementable que servirá como llave primaria
	nombre	text	Nombre con el que se identificará el tipo
	descripción	text	Información adicional que se desea almacenar
	contenedor_hw	boolean	Indica si al componente se le puede asignar hardware
	contenedor_sw	boolean	Indica si al componente se le puede asignar software
Relaciones: --			

Nombre: tarjetas_video_componente			
Descripción: Almacenará la información de las tarjetas de video que pertenecen o han pertenecido a un componente.			
Llave	Campo	Tipo	Descripción
PK	id_tarjeta_video_componente	serial	Auto-incrementable que servirá como llave primaria
FK	id_tarjeta_video	integer	Identificador de la tarjeta de video asociada al componente
FK	id_componente	integer	Identificador del componente que al cual se asoció la tarjeta de video
	fecha_asignado	date	Almacena la fecha en la que se asignó la tarjeta de video al componente
	fecha_cambio	date	Almacena la fecha en la que se sustituyó la tarjeta de video en el componente
	estado	boolean	Indica si está activo o inactivo
	observaciones	text	Información adicional que se desea almacenar
Relaciones: <i>id_tarjeta_video</i> con <i>tarjetas_video</i> <i>id_componente</i> con <i>componentes</i>			

Nombre: tecnicos			
Descripción: Almacenará las personas encargadas de realizar los mantenimientos a los equipos, de las solicitudes realizadas.			
Llave	Campo	Tipo	Descripción
PK	id_tecnico	serial	Auto-incrementable que servirá como llave primaria
	nombre	text	Nombre completo de la persona
	correo	text	Correo institucional de la persona/técnico
	estado	boolean	1 activo, 0 inactivo
	descripcion	text	Información adicional acerca del técnico
Relaciones: --			

Nombre: ubicaciones			
Descripción: Almacenará la información de las áreas, unidades y departamentos en los que se encuentran ubicados los equipos informáticos.			
Llave	Campo	Tipo	Descripción
PK	id_ubicacion	serial	Auto-incrementable que servirá como llave primaria
	nombre	text	Nombre con el que se identifica cada ubicación
	descripción	text	Información adicional sobre la ubicación
Relaciones: --			

Nombre: ups			
Descripción: Almacenará la información sobre las UPS o baterías que se tienen dentro de la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_ups	serial	Auto-incrementable que servirá como llave primaria
FK	id_componente	integer	Identificador del UPS
FK	id_potencia	integer	Identificador de la potencia que tiene el UPS
	estado_bateria	boolean	Campo para determinar si la batería funciona o no
	bateria	boolean	Campo para determinar si el UPS posee batería o no
	regulador	boolean	Especifica si dicho ups solamente se usa como regulador
	ultimo_cambio	date	Última fecha en que se cambió la batería
	observaciones	text	Información adicional que se desea almacenar
Relaciones: <i>id_componente con componentes</i> <i>id_potencia con potencia_ups</i>			

Nombre: usuarios			
Descripción: Almacenará los usuarios que tendrán acceso al sistema.			
Llave	Campo	Tipo	Descripción
PK	id_usuario	serial	Auto-incrementable que servirá como llave primaria
FK	id_ubicacion	integer	Identifica la ubicación a la que pertenecen los usuarios
	nombre	text	Nombre del usuario
	password	text	Contraseña del usuario
Relaciones: <i>id_ubicacion con ubicaciones</i>			

Nombre: velocidad			
Descripción: Almacenará las diferentes velocidades de los dispositivos informáticos que posee la Facultad			
Llave	Campo	Tipo	Descripción
PK	id_velocidad	serial	Auto-incrementable que servirá como llave primaria
	velocidad	text	Almacena el valor de la velocidad
Relaciones: --			

4.5. Diagrama de clases.

La estructura que se muestra a continuación es la utilizada en el proyecto actual, en la cual se utiliza los componentes (EJB) como comunicación con las diversas entidades y clases.

Los EJB se utilizan como contenedor de los métodos principales que ayudan a realizar las funciones como edición, creación, eliminación entre otras, las cuales se han desarrollado de acuerdo con las necesidades de las clases principales, no sin antes heredar y modificar a conveniencia los métodos de la clase AbstractFacade.

Las entidades son las encargadas de la comunicación con la base de datos y de contener los campos que establecen un enlace entre la unidad de persistencia, los EJB y los controladores.

Las clases controladoras son las encargadas de la comunicación entre la capa de negocio y las vistas, conteniendo así los métodos que hacen posibles la interacción del usuario con la aplicación en general.

Un ejemplo de esta comunicación se muestra a continuación, con la tabla Equipos que es la encargada de almacenar la información tal como la ubicación es decir, donde se encuentran físicamente en los diversos departamentos de la facultad y quien es el responsable por dicha máquina.

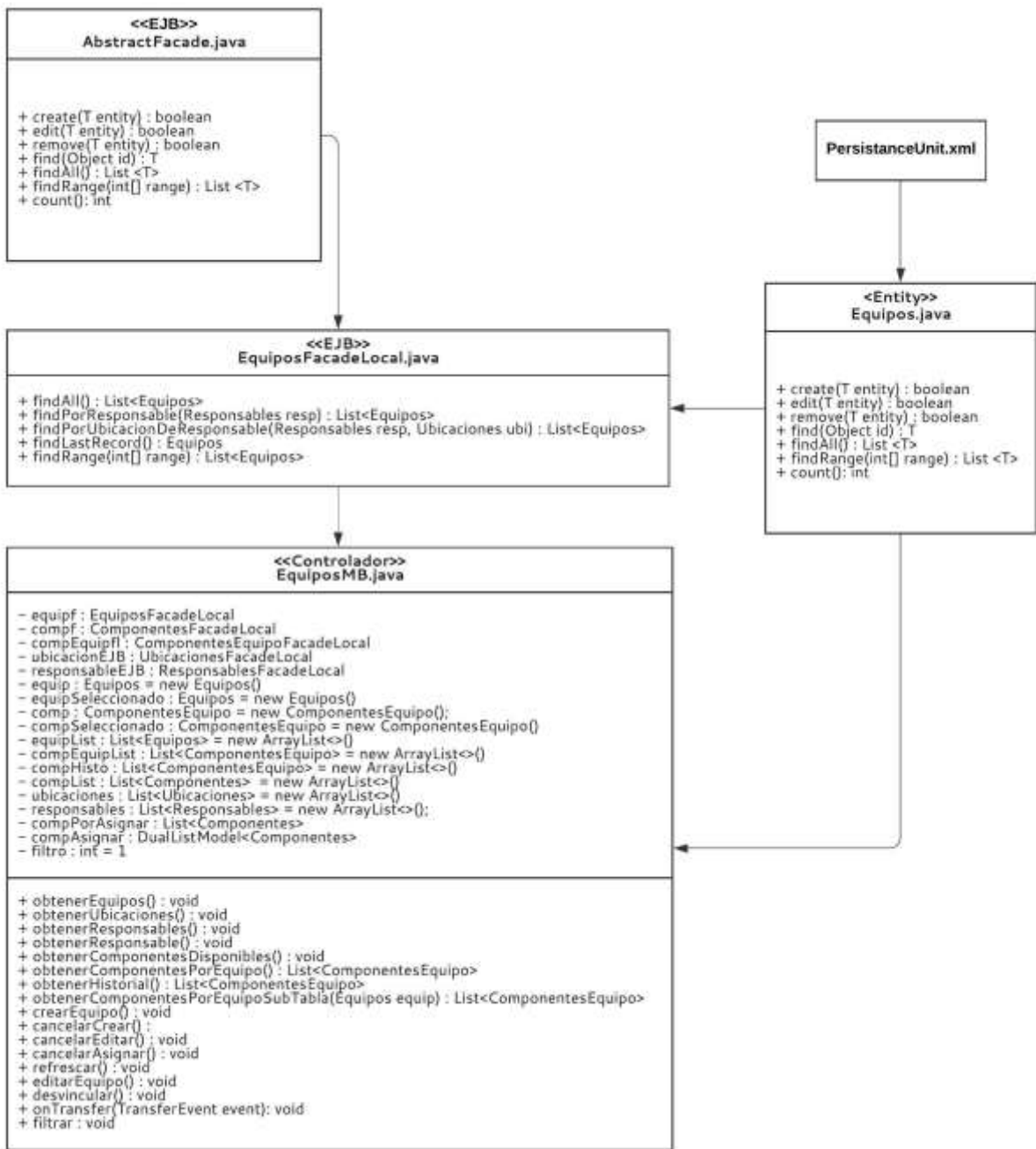


FIGURA 4.4: Diagrama de Clases de Tabla Equipos (Fuente Propia)

4.6. Diagramas de Caso de Uso



FIGURA 4.5: Diagrama general de caso de uso

Nombre	Ver estado	Código: CU-01
Actor(es)	Solicitante	
Descripción	Permite a solicitante consultar el estado del proceso en que se encuentra el servicio de soporte técnico que se solicitó, así como el estado del mantenimiento.	
Precondiciones	El solicitante debe poseer con el código de seguimiento que le fue brindado al momento de realizar la solicitud.	
Poscondiciones		
Flujo de eventos		
Flujo principal	Información requerida	
P1	El solicitante ingresa al sistema.	
P2	El solicitante introduce el código de seguimiento en el campo de comprobación del estado de la solicitud. (E1)	
P3	El sistema presenta al solicitante la información y el estado del proceso en el que se encuentra la solicitud.	
Flujo de excepción		
E1	El solicitante introduce un código no válido.	
E1.1.	El sistema exhibe un mensaje: "No se encontró ningún servicio correspondiente al código ingresado". El sistema retorna al paso P2.	

TABLA 4.14: Modelo caso de uso - Ver estado

Nombre	Notificar solicitud	Código: CU-02
Actor(es)	Solicitante, administrador general del sistema	
Descripción	Notificar al administrador general del sistema cuando una solicitud ha sido emitida.	
Precondiciones	Que al actor haya emitido una solicitud <Emitir solicitud>.	
Poscondiciones		
Flujo de eventos		
Flujo principal	Información requerida	
P1	El solicitante emite una solicitud <Emitir Solicitud>.	
P2	El sistema informa, al administrador general, de solicitud de soporte técnico que ha sido emitida.	

TABLA 4.15: Modelo caso de uso - Notificar solicitud

Nombre	Emitir solicitud	Código: CU-03
Actor(es)	Solicitante, administrador general del sistema	
Descripción	Permite al autor emitir una solicitud de soporte técnico.	
Precondiciones		
Poscondiciones		
Flujo de eventos		

Flujo principal	Información requerida
P1	El actor ingresa al sistema.
P2	El actor introduce los datos en los campos de solicitud. Los datos serían: nombre del responsable del equipo, descripción del problema. (A1)
P3	El actor emite la solicitud. (E1)
Flujo Alternativo	
A1	Registrar el servicio solicitado.
A1.1.	El administrador general introduce los datos para registrar el servicio solicitado. Los datos serían: técnico, prioridad, número de inventario del equipo, observaciones.
A1.2.	El administrador general emite y registra el servicio. (E2)
Flujo de excepción	
E1	Campos vacíos.
E1.1.	El sistema exhibe un mensaje: "Por favor llene los campos obligatorios para emitir la solicitud". El sistema retorna al paso P2.
E2	Campos vacíos.
E2.1.	El sistema exhibe un mensaje: "Por favor llene los campos obligatorios para registrar la solicitud". El sistema retorna al paso A1.

TABLA 4.16: Modelo caso de uso - Emitir solicitud

Nombre	Registra servicio	Código: CU-04
Actor(es)	Administrador general del sistema	
Descripción	Permite al administrador general registrar un servicio de soporte técnico en el sistema.	
Precondiciones	Que una solicitud de servicio técnico haya sido por un actor.	
Poscondiciones		
Flujo de eventos		
Flujo principal	Información requerida	
P1	El actor ingresa al sistema.	
P2	El administrador general introduce los datos para registrar el servicio solicitado. Los datos serían: técnico, prioridad, número de inventario del equipo, observaciones.	
P3	El administrador general emite y registra el servicio. (E1)	
P4	El sistema genera el detalle del servicio de soporte técnico asociado a la solicitud.	
Flujo de excepción		
E1	Campos vacíos.	
E1.1.	El sistema exhibe un mensaje: "Por favor llene los campos obligatorios para emitir la solicitud". El sistema retorna al paso P2.	

TABLA 4.17: Modelo caso de uso - Registrar servicio

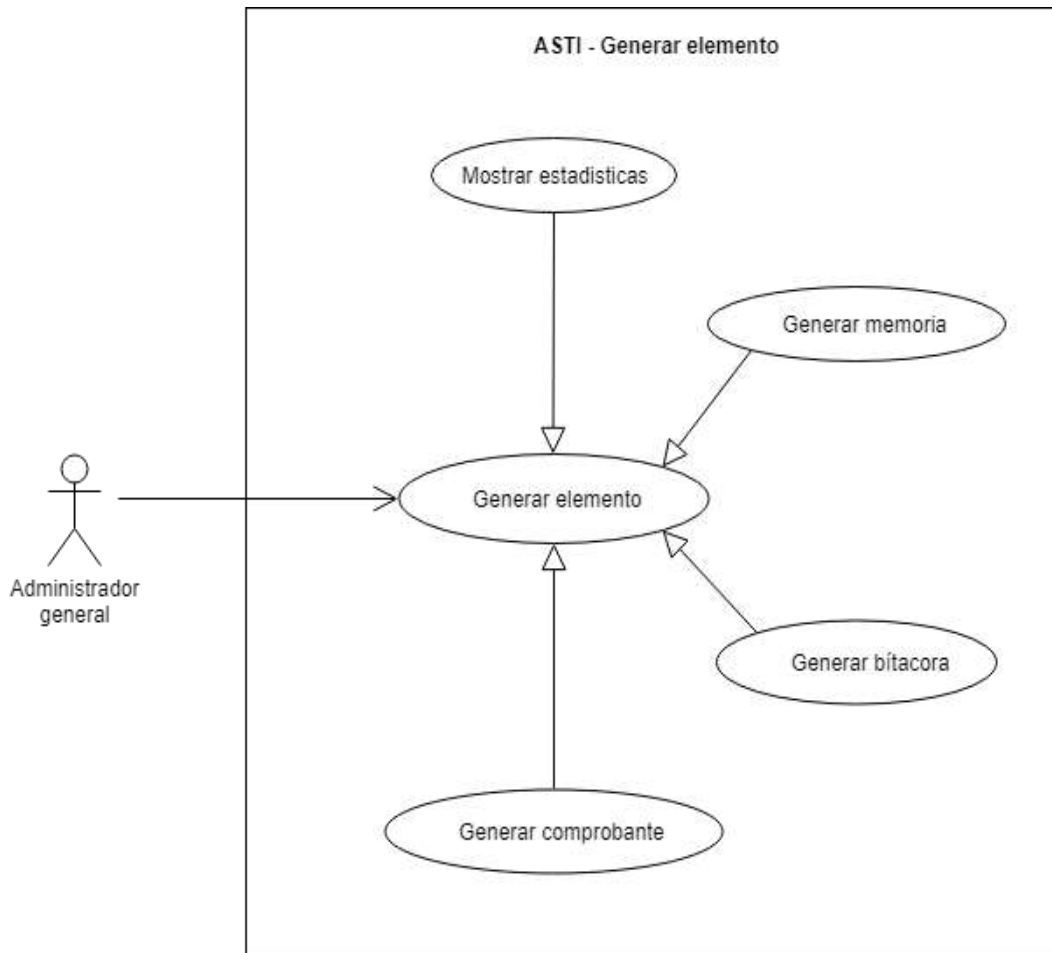


FIGURA 4.6: Diagrama de caso de uso - Generar elemento

Nombre	Generar elemento	Código: CU-05
Actor(es)	Administrador general del sistema	
Descripción	Permite al administrador general generar y mostrar estadísticas, la memoria de servicio o el comprobante de servicio de soporte técnico.	
Precondiciones		
Poscondiciones		
Flujo de eventos		
Flujo principal	Información requerida	
P1	El actor ingresa al sistema, a la sección de su elección.	
P2	En base a los criterios que elija se generará y mostrará las estadísticas, la memoria de servicio o el comprobante de servicio de soporte técnico.	
P3	El sistema exhibirá la información generada.	

TABLA 4.18: Modelo caso de uso - Generar elemento

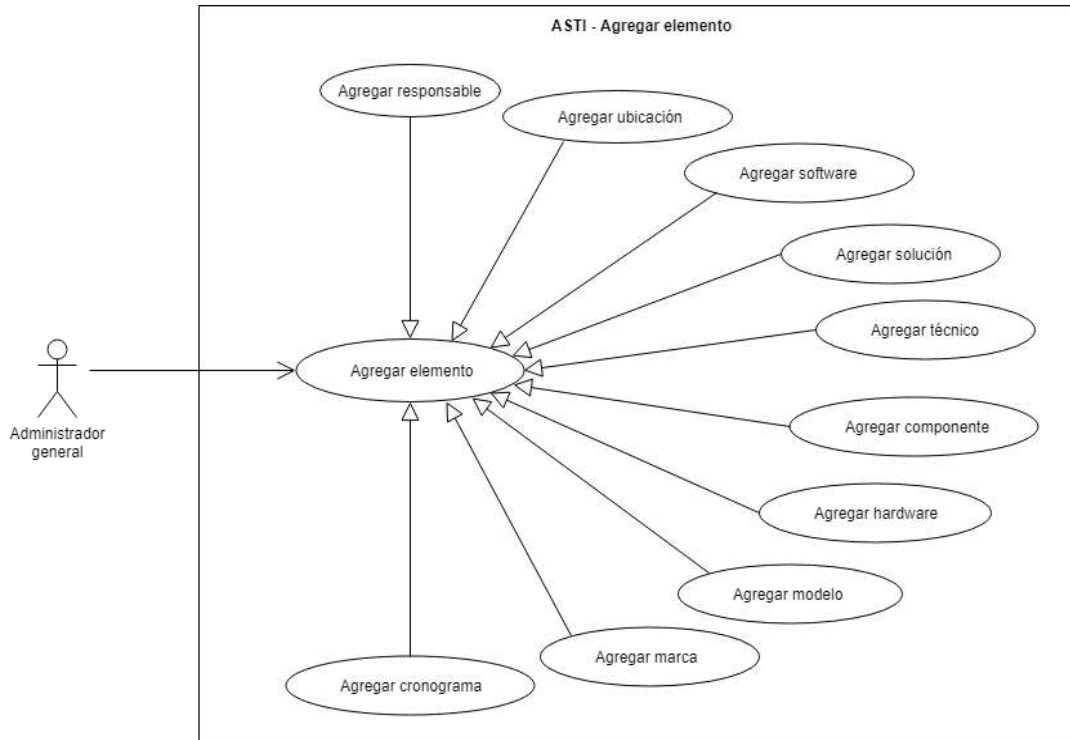


FIGURA 4.7: Diagrama de caso de uso - Agregar elemento

Nombre	Agregar elemento	Código: CU-06
Actor(es)	Administrador general del sistema	
Descripción	Permite al administrador general agregar a la base de datos información sobre nuevos responsables, ubicaciones, técnicos, soluciones, softwares, hardwares, componentes, modelos, marcas o cronogramas, en base a su criterio.	
Precondiciones		
Poscondiciones		
Flujo de eventos		
Flujo principal	Información requerida	
P1	El actor ingresa al sistema, a la sección de su elección.	
P2	En base a los criterios que elija el administrador general se agregará información sobre nuevos responsables, ubicaciones, técnicos, soluciones, softwares, hardwares, componentes, modelos, marcas o cronogramas.	
P3	El sistema almacenará la información proporcionada. (E1)	
Flujo de excepción		
E1	Campos vacíos.	
E1.1.	El sistema exhibe un mensaje: "Por favor llene los campos obligatorios". El sistema retorna al paso P2.	

TABLA 4.19: Modelo caso de uso - Agregar elemento

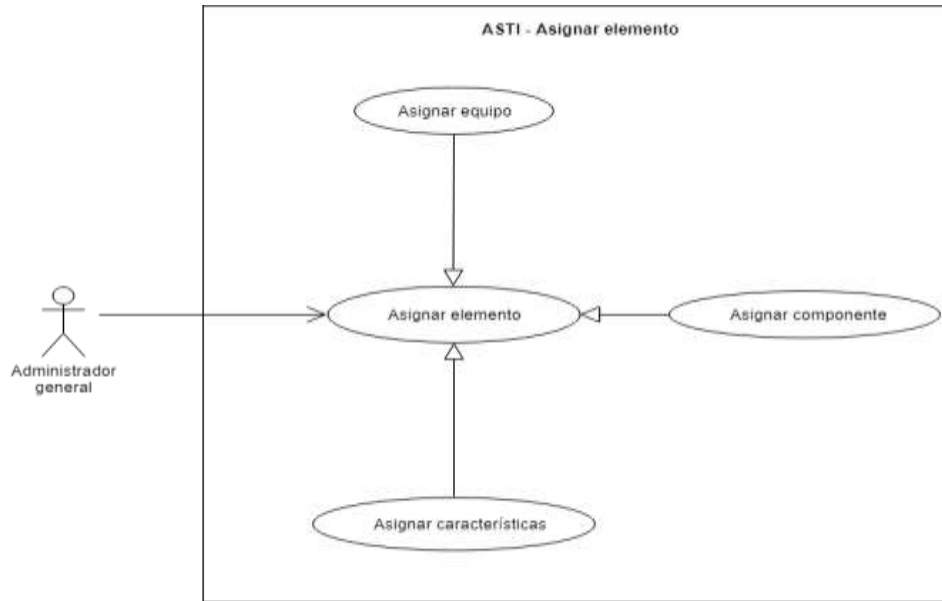


FIGURA 4.8: Diagrama de caso de uso - Asignar elemento

Nombre	Asignar elemento	Código: CU-07
Actor(es)	Administrador general del sistema	
Descripción	Permite al administrador general asignar componentes a un equipo, un equipo a un responsable o asignar un equipo al cronograma de mantenimiento preventivo.	
Precondiciones		
Poscondiciones		
Flujo de eventos		
Flujo principal	Información requerida	
P1	El actor ingresa al sistema, a la sección de su elección.	
P2	En base a los criterios que el administrador general elija se asignan componentes a un equipo, un equipo a un responsable o asignar un equipo al cronograma de mantenimiento preventivo.	
P3	El sistema almacenará la información proporcionada. (E1)	
Flujo de excepción		
E1	Campos vacíos.	
E1.1.	El sistema exhibe un mensaje: "Por favor llene los campos obligatorios". El sistema retorna al paso P2.	

TABLA 4.20: Modelo caso de uso - Ver estado

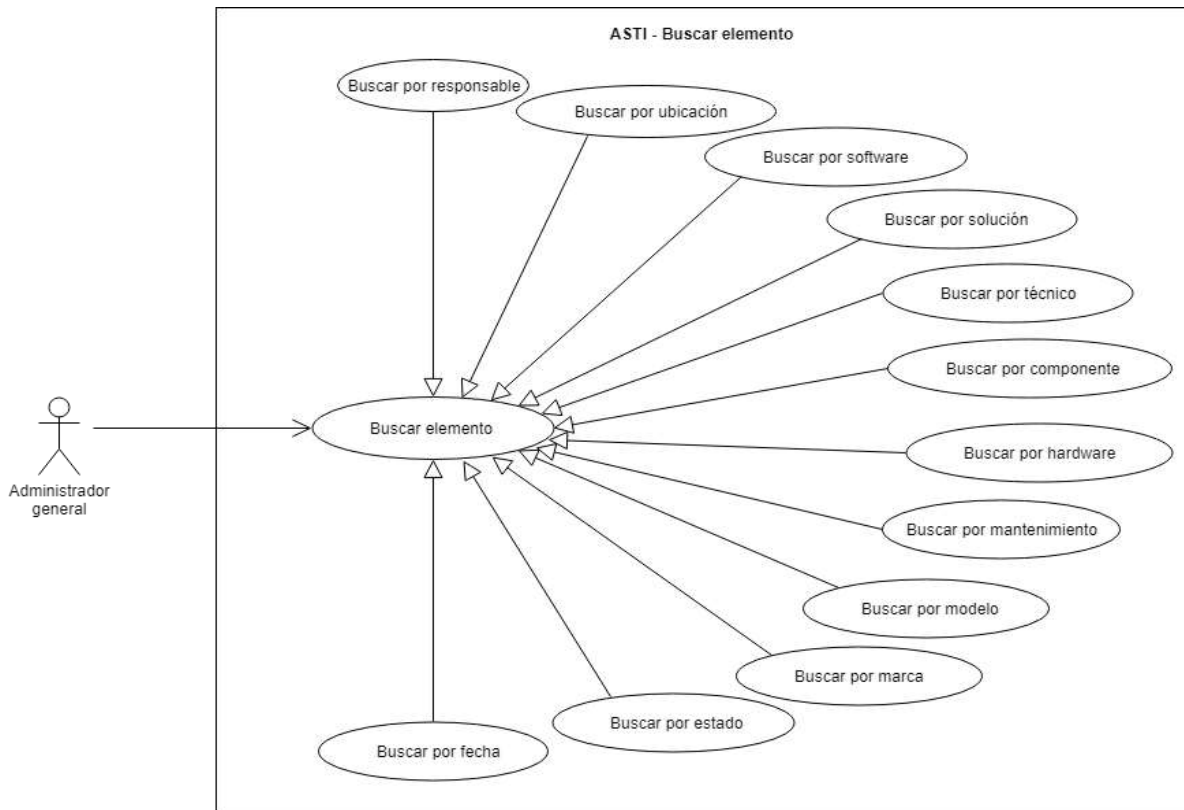


FIGURA 4.9: Diagrama de caso de uso - Buscar elemento

Nombre	Buscar elemento	Código: CU-08
Actor(es)	Administrador general del sistema	
Descripción	Permite al administrador general realizar búsquedas filtradas por responsable, ubicación, técnico, solución, software, hardware, componente, equipo, modelo, marca, estado o fechas.	
Precondiciones		
Poscondiciones		
Flujo de eventos		
Flujo principal	Información requerida	
P1	El actor ingresa al sistema, a la sección de su elección.	
P2	En base a los criterios que el administrador general elija se realizar búsquedas filtradas por responsable, ubicación, técnico, solución, software, hardware, componente, equipo, modelo, marca, estado fechas o una combinación de estas si es posible.	
P3	El sistema exhibirá información en base al filtro de búsqueda. (E1)	

TABLA 4.21: Modelo caso de uso - Buscar elemento

4.7. Metodología de desarrollo.

Se trata del proceso cuya finalidad es desarrollar productos o soluciones para un cliente o mercado en particular, teniendo en cuenta factores como los costes, la planificación, la calidad y las dificultades asociadas. Es decir, se trata del proceso que se suele seguir a la hora de diseñar una solución o un programa específico. Tiene que ver, por tanto, con la comunicación, la manipulación de modelos y el intercambio de información y datos entre las partes involucradas. O para ser más precisos, las metodologías de desarrollo de software son enfoques de carácter estructurado y estratégico que permiten el desarrollo de programas con base a modelos de sistemas, reglas, sugerencias de diseño y guías. (Qué son las metodologías de desarrollo de software. Recuperado de: <https://www.obs-edu.com/int/blog-project-management/metodologia-agile/que-son-las-metodologias-de-desarrollo-de-software>)

Existen diferentes modelos asociados las metodologías de desarrollo del software, cada uno de ellos, tiene un enfoque bien marcado, estos enfoques se utilizan para la planeación y desarrollo de software. Como modelos del ciclo de vida del software tradicionales tenemos:

- Modelo en cascada.
- Modelo de desarrollo evolutivo.
 - o Método de prototipos.
 - o Modelo en espiral.
- Modelo incremental.
- Modelos iterativos.

Como modelos de desarrollo de software en metodologías ágiles tenemos:

- Programación extrema (XP).
- Scrum.
- Kanban.

Las diferencias principales que existen entre metodologías de desarrollo del software tradicional y ágiles tenemos:

Metodologías tradicionales	Metodologías ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.	Basadas en heurísticas provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Especialmente preparados para cambios durante el proyecto.
Proceso mucho más controlado, con numerosas políticas/normas.	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Más roles.	Pocos roles.
Grupos grandes y posiblemente distribuidos.	Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.
La arquitectura del software es esencial y se expresa mediante modelos.	Menos énfasis en la arquitectura del software.
Existe un contrato prefijado.	No existe contrato tradicional o al menos es bastante flexible.

TABLA 4.22: Comparativa entres metodologías de desarrollo

En base a lo mencionado se optará por realizar un ciclo de vida del software basándose en metodologías tradicionales, en específico, en el modelo de desarrollo evolutivo por los siguientes criterios:

- Debido a la disponibilidad del cliente se optará las metodologías tradicionales ya que no podría formar parte del equipo de desarrollo y no podría asistir a reuniones de manera constantemente.
- El modelo de prototipos nos presenta una solución a ese problema, nos acoplamos a su disponibilidad y se le presentará una parte de software funcional, en base a los requerimientos y necesidades, para que pueda evaluarlo. Esto no brindará una visión más clara al equipo de desarrollo y ofrecer un mejor enfoque del producto en las interacciones que tendrá con el usuario final cuando esté terminado así reduciendo el riesgo que este no satisfaga las necesidades del cliente.
- A pesar de que las tecnologías ágiles presentan flexibilidad antes los cambios, la metodología evolutiva ofrece la misma característica, adaptándose a los

cambios de los requisitos del producto. Estas se podrían presentar una vez el cliente evalúe un prototipo. Este factor se adapta a la perspectiva técnico-científico elegida para el desarrollo de la investigación (3.4 Diseño de investigación).

4.7.1. Modelo de desarrollo evolutivo.

El software, como todos los sistemas complejos, evoluciona en el tiempo. Es frecuente que los requerimientos del negocio y del producto cambien conforme avanza el desarrollo, lo que hace que no sea realista trazar una trayectoria rectilínea hacia el producto final; los plazos apretados del mercado hacen que sea imposible la terminación de un software perfecto, pero debe lanzarse una versión limitada a fin de aliviar la presión de la competencia o del negocio; se comprende bien el conjunto de requerimientos o el producto básico, pero los detalles del producto o extensiones del sistema aún están por definirse. En estas situaciones y otras parecidas se necesita un modelo de proceso diseñado explícitamente para adaptarse a un producto que evoluciona con el tiempo (Pressman, R. S. (2010). Ingeniería del software, Un Enfoque Práctico. 7ª Edición. McGraw-Hill).

Para la elaboración del sistema informático se propone trabajar bajo el uso de este modelo y utilizando el método de prototipos como base para el ciclo de vida de desarrollo del software.

4.7.2. Método de prototipos.

Consiste básicamente en que en base a los requerimientos y necesidades que tiene el cliente, se realiza de forma rápida un prototipo, este no vendrá completo ni mucho menos terminado, pero si permitirá contar con las bases necesarias para que cualquier programador pueda seguir trabajando hasta llegar al código final.

4.7.3. Etapas de desarrollo de software para métodos de prototipo.

Plan rápido: A diferencia de otras metodologías, la planeación debe ser muy rápida, en esta fase no se podrá demorar mucho, ya que solamente será un prototipo por el momento.

- Modelado: Nuevamente, una fase que deberá ser suficientemente rápida como para que no lleve nada de tiempo. Hacer el modelado de prototipo será en un corto periodo de tiempo.
- Construcción del prototipo: Ya que cuenta con la planeación de lo que realizará y el modelado rápido, entonces se prosigue con la elaboración del prototipo. Para esta instancia, en contraste con las anteriores, la elaboración del prototipo se realizará de forma prudente y con el tiempo necesario para después mostrarlo al cliente.
- Despliegue: Posterior a contar con el prototipo elaborado y haberlo mostrado al cliente, es momento de comenzar el desarrollo. Este tomará una gran cantidad de tiempo, dependiendo del tamaño del proyecto y el lenguaje de programación que se vaya a utilizar.
- Entrega y Retroalimentación: Una de las cosas con las que cuenta el modelo de prototipos, es que, una vez entregado el proyecto, se deberá dar al cliente cierta retroalimentación sobre cómo lo utilizará.
- Comunicación: Es importante que, una vez entregado el proyecto, se tenga cierta comunicación con el cliente, básicamente para que nos indique si el proyecto es correcto o si desea agregar ciertas funciones, esta metodología lo permite.
- Entrega del producto final: Por último, solamente quedará entregar el sistema elaborado. Aquí se tendrá la ventaja de que el código es reutilizable, para que así con el prototipo ya poder empezar de nuevo y con una buena base de código que acelerará el proceso.

En la *FIGURA 4.8* se muestra la representación del ciclo de desarrollo de software para métodos de prototipo.

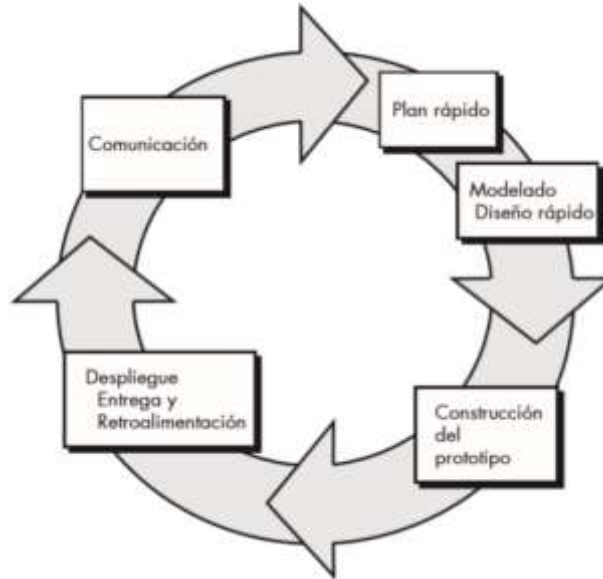


FIGURA 4.10: Ciclo de desarrollo - Método de prototipos
 (Pressman, R. S. (2010). Ingeniería del software, Un Enfoque Práctico. 7ª Edición)

4.8 Estándares de Programación.

Cuando se desarrolla un proyecto informático es importante seguir ciertas normas de codificación ya que facilita la legibilidad del código disminuyendo el tiempo dedicado al mantenimiento y la corrección.

Se utilizarán los siguientes estándares para asegurar la calidad del código:

4.8.1 Nombre de Variables.

- Los nombres de las variables deben ser cortos pero significativos.
- Empezarán por letra minúscula y si se trata de palabras concatenadas, la primera letra de las siguientes palabras en mayúscula.
- No deberán comenzar con un guion bajo ('_') u otros caracteres.
- Serán mnemotécnicos, es decir, diseñado para indicar al observador casual la intención de su uso.
- Se deben evitar los nombres de variable de un carácter, excepto para variables temporales.
- Los nombres comunes para las variables temporales son: i, j, k, m y n para enteros; c, d y e para los caracteres.

Ejemplo:

```
String nombre;
```

```
String ordenesTrabajo;
```

4.8.2 Variables Constantes.

Serán declaradas totalmente en mayúsculas, separando las palabras con el guion bajo “_” si se trata de una palabra compuesta.

Por ejemplo:

```
static final double MAXIMO = 10
```

4.8.3 Métodos.

Los nombres de los métodos serán verbos, con la primera letra de cada palabra interna en mayúscula.

Ejemplo:

```
agregarEquipo ();
```

4.8.4 Clases.

Las clases deberán tener la primera letra mayúscula al igual que la primera letra de las siguientes palabras si se trata de una palabra compuesta. Los identificadores de las clases serán sustantivos.

Ejemplo:

```
class TipoComponente;
```

```
class Modelos;
```

4.8.5 Paquetes.

Los nombres de los paquetes serán palabras separadas por puntos que se almacenan en directorios que coinciden con esos nombres y deberán estar en

minúsculas. El prefijo del nombre del paquete debe ser el nombre del dominio y las siguientes palabras variarán según el acuerdo conveniente.

Ejemplo:

com.edu.uesocc.proyecto

Capítulo V: Conclusiones y Recomendaciones

5.1 Conclusiones.

El desarrollo de un sistema que se adapta a las necesidades de una entidad es necesario, para que el trabajo realizado cuente con las herramientas más oportunas que permitan a los usuarios un mejor desempeño de su labor.

Mediante el uso del Sistema se mejora el control de los mantenimientos preventivos y correctivos de los equipos pertenecientes a la Universidad de El Salvador. Por lo tanto, con la implementación de este se disminuye el trabajo realizado por el Área de Mantenimiento, lo que permite una mejor gestión de su trabajo y libera el tiempo invertido en llevar a cabo la generación de reportes ya que el sistema provee de forma automática información oportuna, confiable y eficiente, mejorando así la calidad de los servicios.

Otro punto importante por destacar ha sido el tener donde Almacenar los datos. Por lo cual, al crear de una base de datos propia para el sistema, se hace posible tener de forma virtual la información, lo que permite la modificación de esta y realizar consultas con mayor velocidad, eliminando así las búsquedas manuales que refieren mayor tiempo y pueden ocasionar errores humanos los cuales entregan información inoportuna e inexacta al momento de requerirse un informe acerca de la labor realizada, además el almacenar permite disponer en cualquier momento de información precisa y al día.

También se desarrolló la solicitud en línea que da un acercamiento más profesional y eficiente, lo que a su vez incorpora un rastreo del estado de las solicitudes, por medio del ingreso del número que se generó cuando se realizó la petición del mantenimiento desde la plataforma en línea, lo que les permite un control con respecto a los detalles de los problemas que poseen los equipos que fueron enviados a revisión o reparación, esto como medida para eliminar la incertidumbre del proceso realizado y ahorrar lo invertido en papel por cada reporte e informe de los problemas.

Finalmente, es importante destacar que el sistema debía poseer una interfaz de usuario amigable e intuitiva, de manera que los solicitantes y los técnicos no tengan que invertir demasiado tiempo aprendiendo a utilizar la herramienta. Es por esto por lo que el sistema cuenta con una serie de pasos que guían al usuario de forma secuencial para realizar las peticiones en línea, lo que hace más fácil su uso, realzando su gestión y aumentando la productividad del equipo, el cual tiene a su cargo múltiples solicitudes de los diversos departamentos de la Facultad.

Con todo lo anterior concluimos que el sistema mejora la manejabilidad de la información necesaria que está presente en el área lo cual es importante para toda la Facultad ya que es un indicador del estado del equipo informático y de las posibles inversiones que esto conlleva para mejorar el ambiente educativo y garantizar siempre el buen funcionamiento de estos.

5.2 Recomendaciones.

Basado en las conclusiones de este proyecto, a continuación, se presentan algunas recomendaciones.

Para el Área de Mantenimiento:

- Se recomienda realizar las adecuaciones necesarias y cumplir con los requisitos de hardware y software para que el sistema pueda ser funcional en la brevedad posible, ya que esto permitirá que su trabajo se simplifique.
- Se sugiere incentivar el uso de las plataformas virtuales, que los técnicos den a conocer a los solicitantes la nueva metodología con la cual se pretende facilitar la creación y continuidad de las solicitudes.
- Innovar procesos, buscar alternativas para que el trabajo del área se distribuya de una manera más eficiente y puedan lograr un método que ayude a simplificar su trabajo.
- Es necesario agendar capacitaciones para todos los usuarios del sistema a implementar, con el fin de promover el correcto uso de este y así evitar futuros problemas tanto del sector solicitante como técnico.
- Realzar la importancia del buen uso, para evitar el deterioro del equipo informático.
- Dar a conocer a la comunidad educativa en general de la Facultad la pronta atención a los problemas de los equipos ya que estos son de vital importancia en el desarrollo y desenvolvimiento de los estudiantes y docentes.

Referencias Bibliográfica

Bibliografía

- Gary S. Walker. (2001). IT Problem Management. Estados Unidos: Prentice Hall.
- Cegarra Sánchez, J. (2004). Metodología de la Investigación Científica y Tecnológica. España: Diaz de Santos.
- Álvarez Gayou. (2003). Diseños del proceso de investigación cualitativa. España: Savin-Baden y Major.
- García Córdoba, F.. (2012). La perspectiva de la investigación tecnológica en educación: una propuesta para promover la calidad. Mexico: Instituto Superior de Ciencias de la Educación del Estado de México.
- Contreras, T., G. y F. Cumsille G. (2003), Metodología de la Investigación Científica. España: Bol. Cient. Asoc. Segur.
- Soto R.,& Berosca, I.. (2012). Investigación científica y tecnológica como factores de Innovación. España: spaciIBC.
- Jones, C.. (1981). métodos de diseño. Reino Unido: Architecture Design and Technology Press.
- Elmasri R.,& Navathe, S.. (2007). Fundamentos de sistemas de bases de datos. Estados Unidos: ADDISON WESLEY.
- Silberschatz, A., Korth, Henry F., & Sudarsha, S.. (2002). Fundamentos de bases de datos. España: McGraw-Hill Inc.

- Duran, F., Gutierrez, F., & Pimentel, E.. (2007). Programación orientada a objetos con Java. España: Thomson Ediciones.

- Perez, M.. (2014). Programacion orientada a los aparatos y programacion estructurada. Estados Unidos: Createspace Independent Publishing Platform.

- ISO/IEC/IEEE. (2010). 830-1998 - IEEE Recommended Practice for Software Requirements Specifications. 8/8/2019, de IEEE Sitio web: <https://ieeexplore.ieee.org/servlet/opac?punumber=5841>

-

- Tarradelles J.. (2019). ¿Qué son las metodologías de desarrollo de software? Agosto 20, 2019, de Business School, Universidad de Barcelona Sitio web: <https://obsbusiness.school/int/blog-project-management/metodologia-agile/que-son-las-metodologias-de-desarrollo-de-software>

ANEXOS

Anexo A

Guía de Observación

1. Concepto de Observación:

La inspección y estudio realizado por el investigador, mediante el empleo de sus propios sentidos, con o sin ayuda de aparatos técnicos, de las cosas o hechos de interés, tal como son o tienen lugar espontáneamente (Sierra y Bravo,1984).

2. Situación o elemento a Observar:

Procesos realizados en el Área de Soporte y Tecnología Informática de la Facultad Multidisciplinaria de Occidente de la Universidad de El Salvador.

3. Objetivo:

Realizar un estimado del tiempo empleado en los procesos de registro de solicitudes, digitalización de información y generación de reportes.

4. Datos Generales:

- Fecha de Observación:

- Lugar de Observación:

- Hora de Observación:

5. Registro de los Datos:

Hora	Actividad