

**INSTITUTO SUPERIOR UNIVERSITARIO**

**“SAN GABRIEL”**



**ESPECIALIDAD INFORMÁTICA**

**PROYECTO DE TITULACIÓN**

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE:**

**TECNÓLOGO EN INFORMÁTICA MENCIÓN ANÁLISIS EN SISTEMAS**

**TEMA:**

DISEÑO E IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA EL REGISTRO DE MANTENIMIENTO DE LOS EQUIPOS DE CÓMPUTO DEL GAD MUNICIPAL DE ALAUSI, DESARROLLADO EN JAVA CON BASE DE DATOS POSTGRESQL EN EL AÑO 2019.

**AUTOR:**

RENE ARMANDO BUÑAY MARCATOMA

**TUTOR:**

ING. JUAN PABLO ROMERO

**RIOBAMBA-ECUADOR**

**JUNIO - 2022**

## **CERTIFICACIÓN**

Certifico que el Sr. Rene Armando Buñay Marcatoma, con N° de Cédula 0605163591 ha elaborado bajo mi Asesoría el Trabajo de Investigación titulado:

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMATICO PARA EL REGISTRO DE MANTENIMIENTO DE LOS EQUIPOS DE CÓMPUTO DEL GAD MUNICIPAL DE ALAUSI, DESARROLLADO EN JAVA CON BASE DE DATOS POSTGRESQL EN EL AÑO 2019.

Por tanto, autorizo la presentación para la calificación respectiva.

---

**Ing. Juan Pablo Romero**

**TUTOR DE TESIS**

“El presente Proyecto de Grado constituye un requisito previo para la obtención del Título de **Tecnólogo en Informática Mención Análisis de Sistemas**”



“Yo, **RENE ARMANDO BUÑAY MARCATOMA**, de cédula de identidad No. 060516359-1, declaro que la investigación es absolutamente original, auténtica, personal y las conclusiones con las que se culmino es de mi absoluta responsabilidad”.

---

Rene Armando Buñay Marcatoma

**APROBACIÓN DE TESIS**

**INSTITUTO TECNOLÓGICO SUPERIOR PARTICULAR “SAN GABRIEL”**

**CARRERA**

**INFORMÁTICA**

**TESIS DE GRADO**

**PREVIA A LA OBTENCIÓN DEL TÍTULO DE:**

**TECNÓLOGO EN INFORMÁTICA MENCIÓN ANÁLISIS DE**

**SISTEMAS**

**TEMA:**

DISEÑO E IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA EL REGISTRO DE MANTENIMIENTO DE LOS EQUIPOS DE CÓMPUTO DEL GAD MUNICIPAL DE ALAUSI, DESARROLLADO EN JAVA CON BASE DE DATOS POSTGRESQL EN EL AÑO 2019.

**APROBADO**

**ASESOR DE TESIS DE GRADO: .....**

**PRESIDENTE DEL TRIBUNAL: .....**

**MIEMBRO DEL TRIBUNAL: .....**

**MIEMBRO DEL TRIBUNAL: .....**

<b>FIRMAS DE RESPONSABILIDAD</b>		
<b>NOMBRE</b>	<b>FIRMA</b>	<b>FECHA</b>
<p>Ing. Juan Pablo Romero</p> <p>DIRECTOR DE TESIS</p>		
<p>MIEMBRO DEL TRIBUNAL</p>		
<p>MIEMBRO DEL TRIBUNAL</p>		
<p>MIEMBRO DEL TRIBUNAL</p>		

**NOTA DE LA TESIS:** .....

## **DEDICATORIA**

Con gratitud y respeto presento el siguiente trabajo, que va especialmente dirigido a mis queridos padres y hermanos por darme su permanente e invaluable apoyo incondicional y consejos para continuar con los estudios superiores.

Rene

## **AGRADECIMIENTO**

Agradezco infinitamente a mi Dios que en todo este tiempo me ha mantenido con salud y vida, ha incrementado mi sabiduría y me ha cuidado durante todo el camino, A mis familias que siempre me estado apoyándome todo momento y a todas las personas que desinteresadamente aportaron para que la propuesta se desarrolle de la mejor manera. A la Ing. Juan Pablo Romero por su apoyo, dirección y guía para el desarrollo y la culminación de mi tesis. A los docentes del Instituto Tecnológico Superior San Gabriel por sus horas de dedicación para impartirnos sus conocimientos profesionales.

Rene

## **ABREVIATURAS**

**ACID:** Atomicidad, Consistencia, Aislamiento y Durabilidad

**API:** Interfaz de programación de aplicaciones

**EN:** Enumeration

**EX:** Exception

**GAD:** Gobierno Autónomo Descentralizado

**GADMCA:** Gobierno Autónomo Descentralizado Municipal del Cantón Alausí

**IDE:** Entorno de desarrollo integrado

**JRE:** Java Runtime

**JVM:** Java Virtual Machine

**MySQL** My Structured Query Language o Lenguaje de Consulta Estructurado

**OB:** Object

**PSF:** Public Static Final

**PSFB:** Public Static Final Boolean

**PSFB:** Public Static Final Int

**PSFI:** Public Static Final String

**SI:** Sistema Informático

**SQL:** Structured Query Language

**UML:** Unified Modeling language

**WWW** World Wide Web



## **GLOSARIO DE TÉRMINOS**

### **Applet**

Es un programa que puede incluirse en páginas web, realizado en un lenguaje de programación denominado Java. Permite de forma sencilla diseñar a los usuarios sus propios recursos didácticos a través de escenas animadas interactivas.

### **Firewall**

Es un guardia de seguridad colocado en el punto de entrada entre una red privada e Internet exterior, de modo que todos los paquetes entrantes y salientes tienen que pasar a través de él. La función de un firewall es examinar cada paquete entrante o saliente y decidir si aceptarlo o descartarlo.

### **Archivo Java (JAR)**

Java es un lenguaje de programación orientado a objetos. El archivo Java (.jar) es un formato de archivo que se emplea para agrupar varios archivos dentro de un único archivo de almacenamiento. Los archivos Java contienen código fuente almacenado en texto sin formato.

### **Java Plugin**

La tecnología Java Plugin forma parte de Java Runtime Environment y amplía la funcionalidad de un navegador web, lo que permite ejecutar applets o Java Beans en el entorno de tiempo de ejecución Java 2 (JRE) de Sun en lugar del entorno de tiempo de ejecución Java que viene con el navegador web.

### **Caché de Java Runtime Environment (JRE)**

Área de almacenamiento de la consola de Java que, en ocasiones, se debe borrar de forma manual para permitir la carga e instalación de la última versión de Java.

### **Java Virtual Machine (JVM)**

Las máquinas virtuales de Java están disponibles para las plataformas de hardware y software de uso más frecuente. En Java, un conjunto de programas de software que permiten la ejecución de instrucciones y que normalmente están escritos en código byte de Java.

### **Javaw.exe**

El archivo javaw.exe es un programa de ejecución que mostrará un cuadro de diálogo durante los momentos en los que se produzca un fallo al iniciar un programa. De esta manera, el proceso javaw.exe resulta ser un programa propiedad de Oracle, Inc., que funciona junto con el explorador Internet Explorer como un plugin de Java.

### **Jucheck.exe**

Se considera un tipo de archivo UpdateChecker Module. Se utiliza más comúnmente en UpdateChecker Module desarrollado por Procedural Arts. Permite a los usuarios ejecutar una variedad de nuevas aplicaciones utilizando varias APIs y brinda soporte para varios tipos de dispositivos de firma digital y más.

### **Manifiesto**

Es un archivo especial que puede abarcar información de los archivos empaquetados en un archivo JAR.

### **Sistema Operativo**

Conjunto de programas que permite manejar la memoria, disco, medios de almacenamiento de información y los diferentes periféricos o recursos de nuestra computadora. Entre los más habituales en su uso se incluyen: Linux, Mac OS X, Microsoft Windows y Solaris.

### **Memoria física**

Este término hace referencia a una forma de almacenamiento semiconductor en equipos conocida como memoria de acceso aleatorio (RAM).

## **Servidor proxy**

Generalmente se trata de un dispositivo u ordenador intermedio que nos permite conectarnos a Internet de manera indirecta. Con ello, es una tecnología que se utiliza como puente entre el origen (un ordenador) y el destino de una solicitud (Internet).

## **Configuración de proxy**

Como norma, los usuarios deben contactar con el administrador de red para obtener la información necesaria para configurar el proxy. Permite que los usuarios se conecten a Internet cuando se utiliza un servidor proxy.

## **JRE privada**

Es el conjunto de elementos que permite diseñar y ejecutar una aplicación de Java, y forma parte del kit de desarrollo. La JRE privada se necesita para ejecutar las herramientas que incluye el JDK.

## **JRE pública**

Se puede eliminar con la opción agregar o quitar programas y también se registra en los exploradores. En este sentido, la JRE pública la pueden emplear otras aplicaciones Java y no va incluida en el JDK.

## **Caché del explorador web**

Serie de archivos como imágenes y otros elementos que el navegador descarga cuando se visita una página web. Para solucionar ciertos problemas de instalación o configuración de Java, a veces es necesario limpiar la caché de forma manual accediendo a un cuadro de diálogo.

## **Solución alternativa**

Se trata de una solución temporal para impedir un problema de sistema identificado a la espera de una solución más permanente.

## **Fecha de caducidad de Java**

Java caduca cada vez que hay disponible una nueva versión con correcciones a las vulnerabilidades de seguridad. Para los sistemas que no se pueden ejecutar en servidores Oracle, un mecanismo secundario se encargará de caducar esta versión de JRE.

## **Línea base de seguridad**

Es la mínima actualización recomendada para Java. No se recomienda a los usuarios que ejecuten versiones de Java que están por debajo de la última línea base de seguridad. Cada versión principal de Java (como Java 6 o Java 7) debe poseer su propia actualización de línea base de seguridad.

## **Alerta de seguridad**

Oracle emitirá una alerta de seguridad (por ejemplo, la versión de una corrección de seguridad fuera del programa CPU normal) en casos en los que la urgencia de una solución requiere que se publique antes de la siguiente actualización de parche crítico.

## **Aplicaciones firmadas y sin firmar**

La firma de código es una tecnología de seguridad por medio de la cual las aplicaciones se firman digitalmente para confirmar el autor o publicador del software. Con aplicaciones sin firmar se hace referencia al código que no ha sido firmado, en cuyo caso el origen de la aplicación no se puede verificar.

# ÍNDICE GENERAL

CERTIFICACIÓN.....	II
DEDICATORIA.....	VI
AGRADECIMIENTO.....	VII
ABREVIATURAS.....	VIII
GLOSARIO DE TÉRMINOS.....	IX
ÍNDICE GENERAL.....	13
ÍNDICE DE TABLAS.....	17
ÍNDICE DE GRÁFICOS.....	18
RESUMEN.....	22
SUMMARY.....	23
CAPÍTULO I.....	24
1. MARCO REFERENCIAL.....	24
1.1 Antecedentes.....	25
1.2 Definición del Problema.....	26
1.3 Justificación.....	26
1.4 Objetivos.....	28
CAPÍTULO II.....	29
MARCO TEÓRICO.....	29
2.1 Sistema Informático.....	30
2.1.1 Definición.....	30
2.1.2 Estructura y funciones de un sistema informático.....	31
2.1.3 Importancia del sistema informático.....	32
2.2 Tipos De Sistemas Informáticos.....	33
2.2.1 Sistema de información gerencial.....	33
2.2.2 Sistema transaccional.....	33
2.2.3 Sistema de apoyo a las tomas de decisiones.....	33
2.2.4 Estratégicos.....	34
2.3 Java.....	34
2.3.1 Historia De Java.....	34
2.3.2 Api y Servicios.....	35
2.3.3 Applets.....	36

2.3.4	Servlets .....	36
2.4	Servidor De Aplicaciones Glassfish .....	37
2.4.1	GlassFish .....	37
2.5	Programación .....	37
2.5.1	Palabras reservadas .....	38
2.5.2	Comentarios en el código .....	38
2.5.3	Empaquetado de objetos .....	38
2.5.4	Definición de paquete .....	39
2.5.5	Sentencias de importación .....	40
2.5.6	Declaración de clase .....	40
2.5.7	Variables .....	41
2.5.8	Métodos .....	41
2.5.9	Sintaxis de la declaración del constructor. ....	41
2.5.10	Métodos estáticos y de instancia .....	41
2.5.11	Adición de variables de clases .....	41
2.5.12	Tipos de datos primitivos .....	42
2.5.13	Cadenas y operadores .....	42
2.5.14	Cadenas .....	42
2.5.15	Operadores .....	43
2.5.16	Sentencia de control .....	45
2.6	Otros Lenguajes De Programación .....	46
2.6.1	Lenguaje simple .....	46
2.6.2	Lenguaje orientado a objetos .....	46
2.6.3	Aplicaciones distribuidas .....	47
2.6.4	Interpretado y compilado .....	47
2.7	Arquitectura .....	47
2.7.1	Componentes .....	48
2.7.2	PostgreSQL .....	48
2.7.3	Características .....	48
2.7.4	Ventajas .....	49
2.7.5	PgAdmin .....	50
2.7.6	NetBeans .....	50
2.7.6.1	Introducción .....	50
2.7.6.2	Desarrollador Java NetBeans .....	51
2.7.6.3	Estructura de la programación Java .....	51

2.7.6.4	Programación detallada en Java.....	52
2.7.6.5	Ventajas .....	52
2.7.6.6	Que contiene un proyecto en NetBeans .....	52
2.7.6.7	Compatibilidad con Java y MySQL.....	53
CAPÍTULO III .....		54
3. ANÁLISIS Y DISEÑO DEL SISTEMA.....		54
3.1	RECOPIACIÓN DE INFORMACIÓN.....	55
3.2	ANÁLISIS .....	56
3.2.1	Estudio de Factibilidad .....	56
3.2.1.1	Factibilidad operativa .....	57
3.2.1.2	Factibilidad técnica .....	58
3.2.1.3	Factibilidad económica .....	60
3.2.1.4	Factibilidad legal.....	61
3.2.2	Análisis de Requerimientos .....	61
3.2.2.1	Requerimientos funcionales del sistema.....	62
3.2.2.2	Requerimientos no funcionales del sistema.....	62
3.2.3	Definición de Casos de Uso.....	63
3.2.3.1	Actores .....	63
3.2.3.2	Login .....	63
3.2.3.3	Registro de datos en el sistema .....	64
3.2.3.4	Registro de mantenimiento .....	65
3.2.3.5	Eliminación y edición de datos .....	66
3.2.3.6	Consulta de datos .....	67
3.2.3.7	Reportes de mantenimientos .....	68
3.3	DISEÑO.....	70
3.3.1	Diseño Conceptual.....	70
3.3.2	Modelo Relacional.....	71
3.3.3	Diccionario de Datos .....	72
CAPÍTULO IV .....		76
4. IMPLEMENTACIÓN DEL SISTEMA .....		76
4.1	Configuración de las herramientas de desarrollo .....	77
4.1.1	PostgreSQL.....	77
4.1.2	NetBeans.....	83
4.2	Arquitectura del Sistema .....	85
4.3	Implementación del Sistema.....	86

4.3.1 Definición de Módulos .....	86
4.3.2 Desarrollo de la aplicación .....	87
4.4 Pruebas de la aplicación .....	87
4.5 Capacitación al Personal.....	88
4.6 MANTENIMIENTO .....	88
CAPÍTULO V .....	89
5. CONCLUSIONES Y RECOMENDACIONES .....	89
5.1 Conclusiones.....	90
5.2. Recomendaciones .....	91
BIBLIOGRAFÍA .....	92
ANEXOS .....	98
MANUAL DEL USUARIO .....	99
MANUAL DEL TÉCNICO .....	108



## ÍNDICE DE TABLAS

<b>Tabla 1</b> Tipos de datos primitivos de Java.....	42
<b>Tabla 2</b> Operadores lógicos .....	44
<b>Tabla 3</b> Operadores aritméticos, relacionales y condicionales del lenguaje Java .....	44
<b>Tabla 4</b> Recursos humanos .....	57
<b>Tabla 5</b> Software.....	59
<b>Tabla 6</b> Hardware .....	60
<b>Tabla 7</b> Gastos económicos .....	61
<b>Tabla 8</b> Mantenimiento.....	72
<b>Tabla 9</b> Usuarios.....	72
<b>Tabla 10</b> Registro de Equipo .....	73
<b>Tabla 11</b> Pruebas en el sistema.....	88

## ÍNDICE DE GRÁFICOS

<b>Gráfico 1</b> Estructura del Sistema Central .....	32
<b>Gráfico 2</b> Entorno desarrollador abierto para algunos sistemas operativos .....	51
<b>Gráfico 3</b> Estructura básica detallada en java.....	52
<b>Gráfico 4</b> Metodología de desarrollo.....	55
<b>Gráfico 5.</b> Ingreso al Sistema.....	64
<b>Gráfico 6.</b> Registro de datos en el sistema.....	65
<b>Gráfico 7</b> Registro de Mantenimiento .....	66
<b>Gráfico 8</b> Eliminación y Edición de Datos .....	67
<b>Gráfico 9</b> Consulta de datos.....	68
<b>Gráfico 10</b> Reportes Impresos .....	69
<b>Gráfico 11</b> Diseño Conceptual.....	70
<b>Gráfico 12</b> Modelo relacional.....	71
<b>Gráfico 13</b> Pantalla de Login.....	73
<b>Gráfico 14</b> Pantalla Principal.....	74
<b>Gráfico 15</b> Asignación de mantenimiento .....	74
<b>Gráfico 16</b> Reportes de mantenimiento .....	75
<b>Gráfico 17</b> Descarga de PostgreSQL.....	78
<b>Gráfico 18</b> Instalación de PostgreSQL .....	78
<b>Gráfico 19</b> Instalación Directory de PostgreSQL.....	79
<b>Gráfico 20</b> Selección de Componentes de PostgreSQL .....	79
<b>Gráfico 21</b> Password de PostgreSQL .....	80
<b>Gráfico 22</b> Puerto de PostgreSQL .....	80
<b>Gráfico 23</b> Stack Builder de PostgreSQL.....	81
<b>Gráfico 24</b> Fuente Propio Iniciar PostgreSQL .....	81
<b>Gráfico 25</b> pgAdmin de PostgreSQL.....	82
<b>Gráfico 26</b> Servidores de PostgreSQL.....	82
<b>Gráfico 27</b> Contraseña de ingreso de PostgreSQL .....	83
<b>Gráfico 28</b> Página de descarga de Neat Beans .....	83
<b>Gráfico 29</b> Ejecución de Netbeans .....	83
<b>Gráfico 30</b> Ubicación de Netbeans .....	84

<b>Gráfico 31</b> Instalación de Netbeans .....	84
<b>Gráfico 32</b> Netbeans 8.2 .....	84
<b>Gráfico 33</b> Arquitectura Cliente – Servidor.....	86
<b>Gráfico 34</b> Definición de módulos del sistema.....	86

## INTRODUCCIÓN

En la actualidad, las Tecnologías de la Información y la Comunicación (TIC's) han tomado un papel importantísimo en la sociedad y se utilizan en multitud de actividades. Por ende, se tratan de programas y herramientas que emiten cierta información a través del uso de soportes tecnológicos.

Con ello, los controles de mantenimiento permiten una herramienta para determinar el estado de las instalaciones, bienes y componentes informáticos, al ser un elemento importante de las labores de prevención. Todas sus componentes y funciones son inspeccionados por parte del jefe de área, de tal forma que su incumplimiento de brindar un buen mantenimiento puede ocasionar pérdidas de recursos y tiempo en cada una de las actividades que requieren de un equipo informático.

En este contexto, es importante manifestar que el Gobierno Autónomo Descentralizado (GADM) del cantón Alausí que la manera de llevar un inventario de los equipos existentes se lo ha realizado de forma manual empleando documentos de texto u hojas de cálculo. Este manejo de los equipos ha ocasionado lentitud en la búsqueda de los trámites necesarios, pérdida de la información y en varios momentos la pérdida de algún equipo que se encontraba habilitado (se lo da por perdido, inhabilitado o se le da de baja).

Ante lo expuesto, la presente investigación se desarrolló con la necesidad de brindar una automatización, inventario y registro de mantenimiento de los equipos informáticos del GADM Alausí. Esta automatización contribuye al ingreso ordenado equipos existentes, ubicación, pertenencia y disponibilidad efectiva en cualquier momento que se necesite.

Para el desarrollo del estudio se aplicó el análisis, diseño y programación, conocimientos adquiridos dentro de la carrera de Tecnología en Informática Mención Análisis de Sistemas, como también los conocimientos obtenidos en el lenguaje de programación JAVA.

Resulta importante mencionar que el objetivo del trabajo de investigación fue diseñar e implementar un sistema informático para el registro de mantenimiento de los equipos de cómputo de GADM del cantón Alausí, a través del desarrollo en Java con base de datos PostgreSQL. A continuación, se exponen la división de capítulos:

El Capítulo I, refiere al marco referencial que abarca la formulación, los antecedentes y la definición del problema, asimismo, contempla la justificación y los objetivos propuestos en el estudio.

El capítulo II, abarca el marco teórico, en este apartado se investigan los conceptos necesarios para el desarrollo del aplicativo, se detalla el lenguaje JAVA el cual me permite realizar la programación y el gestor de base de datos PostgreSQL donde se efectúa el almacenamiento de la información.

El capítulo III, contempla el análisis, diseño del sistema de mantenimiento en el cual se analizan las distintas técnicas que se aplicaron para el desarrollo del estudio, la factibilidad, el análisis de requerimientos funcionales, no funcionales y además del diseño de la arquitectura de la base de datos.

El capítulo IV, este apartado abarca la implementación del sistema, ítem donde se describen con detalle las múltiples herramientas utilizadas para la implementación de la aplicación, arquitectura del sistema y pruebas (validaciones, códigos, redundancia, formatos y reportes).

El capítulo V, se muestran las conclusiones y recomendaciones enfocadas al tema de estudio. Posteriormente, se presenta el manual técnico que desarrolla las herramientas necesarias para el funcionamiento del sistema y el manual de usuario que detalla cada una de las secciones que el sistema presenta.

## RESUMEN

La presente investigación se denomina: “Diseño e implementación de un sistema informático para el registro de mantenimiento de los equipos de cómputo del GADM de Alausí, desarrollado en java con base de datos PostgreSQL en el año 2019”. Mismo que tiene como objetivo de una forma ágil y oportuna mantener el registro efectivo de mantenimiento de los equipos de cómputo que se realizan en la Unidad de TICs. Para el desarrollo de la aplicación se empleó una metodología en cascada aplicando una arquitectura cliente servidor, utilizando el lenguaje de programación JAVA (versión 8.2) que permitió crear formularios, contenidos, mediante el uso de IDEs y herramientas que ayudaron a la creación de las pantallas que los usuarios podrán ejecutar en el momento de cargar la información necesaria al sistema. A su vez, se aplicó un gestor de base de datos PostgreSQL (versión 11.9) que permitió la facilidad de almacenamiento en el motor de Base de Datos, donde se ingresaron: usuarios, equipos, accesorios y el responsable de la administración del sistema, todo esto contribuyó a la realización de tareas como: insertar, modificar, actualización, eliminar y consultar información, finalizando con las pruebas de validación de usuarios nombres y claves, mantenimientos por equipo y usuario, mantenimientos por fechas de ingreso y de entrega, reportes de mantenimientos por usuario y por equipo. Los resultados revelan que con la aplicación web se puede imprimir reportes como: especificación de los equipos que ingresaron a mantenimiento, la fecha de ingreso, la reparación y entrega de los equipos informáticos que fueron ingresados anteriormente, las novedades que presentan los equipos cuando se registran para que se realice el respectivo mantenimiento, saber quién es el custodio del equipo y a que unidad pertenece, el nombre de quien realizó el mantenimiento, las observaciones del equipo y el tiempo en el cual se realizará el siguiente mantenimiento.

## SUMMARY

This research is called: "Design and implementation of a informatics system for the maintenance record of the computer equipment of the GADM of Alausí, developed in java with a PostgreSQL database in the year 2019". The same that has as an agile and timely objective to maintain the effective record of maintenance of the computer equipment that is carried out in the ICT Unit. For the development of the application, a cascade methodology was used, applying a client-server architecture, using the JAVA programming language (version 8.2) that allowed the creation of forms, contents, through the use of IDEs and tools that helped to create the screens. That the users will be able to execute now of uploading the necessary information to the system. In turn, a PostgreSQL database manager (version 11.9) was applied, which allowed easy storage in the Database engine, where users, equipment, accessories and the person responsible for system administration were entered, all this contributed to the performance of tasks such as: inserting, modifying, updating, deleting and consulting information, ending with the validation tests of user names and passwords, maintenance by equipment and user, maintenance by entry and delivery dates, maintenance reports per user and per team. The results reveal that with the web application it is possible to print reports such as: specification of the equipment that entered maintenance, the date of entry, the repair and delivery of the computer equipment that was previously entered, the news that the equipment presents when they are registered. so that the respective maintenance is carried out, knowing who is the custodian of the equipment and to which unit it belongs, the name of the person who performed the maintenance, the observations of the equipment and the time in which the next maintenance will be carried out.

**CAPÍTULO I**  
**MARCO REFERENCIAL**



## **FORMULACIÓN GENERAL DEL PROYECTO DE TESIS**

### **1.1 Antecedentes**

En el GADM del cantón Alausí existen múltiples equipos de cómputo los cuales reciben mantenimiento periódicamente por parte del área de tecnología de la institución, resultando necesario contar con un registro detallado de los mantenimientos ejecutados y requeridos por cada uno de los equipos, con la finalidad de contar con datos estadísticos que permitan realizar futuros arreglos en los equipos.

En Ecuador dentro de las entidades públicas por reglamento institucional resulta necesario contar con un registro de inventario tecnológicos absoluto y detallado, puesto que los equipos de cómputo son de propiedad del estado, por lo que las personas encargadas de los equipos en entidades públicas no pueden dar de baja un equipo sin tener un informe técnico de los profesionales del área de sistemas que lo sustente, esto se debe a que el informe se lo desarrolla con el sustento histórico de mantenimiento que evidencie el deterioro natural de cualquier equipo.

En la actualidad, cada equipo de cómputo existente en la institución el registro de mantenimiento es anotado en libros de registro, mismos que se encuentran en el departamento de tecnología. Estos libros apoyan a llevar un control en el historial de cada uno de los equipos. No obstante, estos registros manuales a través de libros de control se encuentran expuestos a sufrir un deterioro y pérdida de información siendo de vital importancia la creación de un sistema informático de respaldo. No obstante, se muestra otro problema en la tardanza en la realización de reportes, esto debido al retraso en la búsqueda de información y su posterior digitación.

#### **1.1.2 Datos Informativos**

El GADM del cantón Alausí se alinea con su misión y se sustenta en la filosofía y enfoque de productos, servicios y procesos, con el propósito de asegurar su ordenamiento orgánico y un servicio de calidad, descritos como:

#### **Misión**

Constituirse para el año 2023 en un GAD con un modelo de gestión, técnica, participativa y operativa que fundamenta su accionar en el bienestar de la comunidad a

través de un proceso de mejoramiento y tecnificación de eficiencia y eficacia del servicio, mediante una planificación estratégica integral que garantice el progreso (GADM Alausí, 2022).

## **Visión**

Somos una entidad gubernamental encaminada al servicio público local que promueve el desarrollo ordenado y sostenible del cantón y el bienestar de la comunidad de manera eficiente, honesta y responsable, involucrando la participación ciudadana por el bienestar común (GADM Alausí, 2022).

### **1.2 Definición del Problema**

El Gobierno Autónomo Descentralizado del cantón Alausí a través de la Unidad de Tecnología de Información y Comunicación realiza el registro de mantenimiento de los equipos de cómputo existentes en la institución de forma manual empleando documentos de texto u hojas de cálculo que luego de ser impresos son almacenados en archivadores. El origen de este problema radica en la gran cantidad de documentos que se genera al realizar el mantenimiento de los equipos de cómputo, lo que conlleva a un aumento considerable de tiempo en lo que respecta al proceso de elaboración de los distintos informes, lo que no permite un flujo eficaz de información ni de control ni de historial de cada uno de los equipos y al ser manual no cuenta tampoco con un mecanismo de filtrado y búsqueda adecuada para el procesamiento de una gran cantidad de registros del mantenimiento de los equipos de cómputo.

### **1.3 Justificación**

La implementación del Sistema Informático en el GADM del cantón Alausí permitirá la automatización del registro de mantenimiento de sus equipos de cómputo. Esta automatización contribuirá al ingreso ordenado de los equipos existentes, ubicación, pertenencia y disponibilidad efectiva en cualquier momento que se necesite, mecanismo que estará soportado por un motor de base de datos.

Este proyecto es factible, ya que las autoridades del GADM del cantón Alausí y la Unidad de Tecnología de Información y Comunicación facilitaron toda la recopilación de información, así como también el levantamiento de los procesos para desarrollar el

sistema web para el registro mantenimiento de los equipos de cómputo, además el investigador y desarrollador del proyecto tiene todo el conocimiento para la implementación exitosa del proyecto de investigación.

Los beneficiarios del Sistema Web de registro de mantenimientos a equipos de cómputo serán el personal administrativo y tecnológico del GADM del cantón Alausi, ya que el acceso a la información estará disponible en cualquier momento que la institución lo requiera, esto realmente reducirá tiempos, procesos, costos y contribuirá al desarrollo del GAM del cantón Alausi.

## **1.4 Objetivos**

### **1.4.1 Objetivo General**

Diseñar e implementar un sistema informático para el registro de mantenimiento de los equipos de cómputo de GAD Municipal del cantón Alausí desarrollado en Java con base de datos PostgreSQL en el año 2019.

### **1.4.2 Objetivos Específicos**

- Investigar las ventajas que posee el lenguaje de programación Java y la interconexión con el gestor de base de datos PostgreSQL.
- Analizar el proceso de mantenimiento y control de los equipos de cómputo en función a los requerimientos de los procesos internos y de la normativa del GADM de Alausí.
- Diseñar el sistema web para el registro de mantenimiento de los equipos de cómputo en base a los requerimientos funcionales y a la arquitectura de la base de datos.
- Implementar el sistema web para el registro de mantenimiento de los equipos de cómputo de GADM del cantón Alausí utilizando el lenguaje de programación Java y base de datos PostgreSQL.
- Realizar las pruebas de funcionamiento y optimización del sistema automático de registro de mantenimiento de los equipos de cómputo del GAD Municipal del cantón Alausí.

**CAPÍTULO II**  
**MARCO TEÓRICO**

## **2.1 Sistema Informático**

### **2.1.1 Definición**

Un sistema informático es una colección de entidades (hardware, software y liveware) que están diseñadas para recibir, procesar, administrar y presentar información en un formato significativo (White et al., 2017). El objetivo es permitir el tratamiento de manera automática de los datos, un ejemplo es uno o varios ordenadores personales junto a la persona que lo maneja, los programas que contiene y los periféricos que lo envuelven (impresora, teclado, altavoces) (Denby & Lucia, 2020).

En este sentido, Greenbaum & Kyng (2020) mencionan que los componentes del sistema informático: i) hardware informático: son partes físicas/partes intangibles de una computadora. ii) software informático: también conocido como programas o aplicaciones. iii) liveware: es el usuario de la computadora.

En las entidades públicas o privadas los sistemas informáticos son importantes porque ayudan a clasificar, organizar, buscar y puede almacenar cantidades masivas de información (Habrát et al., 2017). Con ello, los mismos presentan cuatro actividades básicas según Gissela (2018) que se describen a continuación:

- Entrada de información: el sistema elige la información a ser procesada por medio del teclado, código de barras, disco compacto CD, entre otros elementos.
- Almacenamiento de información: es la parte del sistema que mantiene los datos accesibles para los procesadores de información. En otras palabras, la unidad de almacenamiento de información de un sistema es un disco duro o un servidor que generalmente contiene una base de datos.
- Procesamiento de la información: es la manipulación de datos para producir información útil; implica la captura de información en un formato que sea recuperable y analizable. Procesar información implica tomar información en bruto y hacerla más útil al ponerla en contexto. En general, el procesamiento de información significa procesar nuevos datos, lo que incluye una serie de pasos: adquirir, ingresar, validar, manipular, almacenar, enviar, comunicar, recuperar y desechar.

- Salida de la información: es la capacidad del sistema de información, para extraer la información almacenada. Por medio de unidades típicas de salida: impresoras, Universal Serial Bus (USB), CD, otras.

Un sistema informático se encuentra conformado por recursos que se explican a continuación:

- Recurso de hardware: elementos como estructuras físicas de redes de ordenadores, impresoras, memorias, escáneres, códigos de barras, etc.
- Recurso de software: elementos como documentos, manuales de usuario, archivos, aplicaciones, bases de datos, sistemas operativos, etc.
- Recurso humano: son las personas que conforman el sistema, pueden ser los operadores, los técnicos o los usuarios finales.

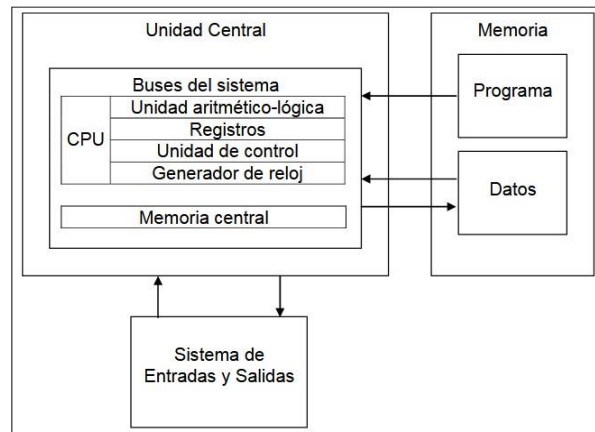
Según Hernández (2018), un sistema informático conlleva varios objetivos descritos a continuación:

- Adaptarse a las necesidades solicitadas por la institución, de manera es éste pueda responder a su constante evolución conforme vaya creciendo y evolucionando, debido que en algunos casos surgen nuevas necesidades que han de ser satisfechas por el sistema.
- Apoyar con los objetivos y estrategias a través de sus funcionalidades, creando información necesaria para que instituciones tengan un correcto funcionamiento.
- Suministrar información de cada una de las actividades ejecutadas por la empresa.
- Tener una buena interacción con cada uno de los actores para los cuales se destina el sistema, de modo que se satisfagan sus necesidades de modo eficaz.

### **2.1.2 Estructura y funciones de un sistema informático**

Dos componentes conforman un sistema informático según Chacon (2017):

**El primero es un sistema central:** permite el proceso de la información por medio de dispositivos periféricos de entrada y salida de información.



**Gráfico 1 Estructura del Sistema Central**

**Fuente:** (Chacon, 2017)

**El segundo componente es el sistema informático:** ayuda a la comunicación entre el sistema central y el usuario denominado Software Informático. Este realiza el proceso de información de entrada y entrega la información de salida (Samantha, 2018, pág. 25).

### 2.1.3 Importancia del sistema informático

Los sistemas informáticos se han convertido en una parte esencial de la vida y tienen un impacto significativo en la sociedad moderna (Cahan & Cimino, 2017). Los sistemas informáticos se encuentran conformados por múltiples componentes de hardware, mismos que se complementan y de componentes de programas que son ejecutados en las computadoras. En las oficinas existen todavía trabajadores que por la mañana preparan su escritorio (Kontopantelis et al., 2018). De esta manera, Roffel & Chin (2017) mencionan que es importante mencionar que lleva mucho tiempo y dinero para las organizaciones manejar información de manera manual. Por lo cual, hoy en día las empresas han optado por utilizar sistemas informáticos para automatizar tareas, con ello han logrado reducir costos, minimizar tiempos, entre otros beneficios (Chacon, 2017). Son múltiples las ventajas que un sistema informático presenta (Katona & Kovari, 2018), algunas se presentan a continuación:

- Automatizan tareas operativas en las organizaciones.
- Apoya a que los procesos y cálculos realizados dentro de las empresas sean precisos, rápidos y simples.
- Por su diseño puede almacenar gran cantidad de información generada dentro de la organización.



- Son sistemas 100% impresentables en una organización (Samantha, 2018).

## **2.2 Tipos De Sistemas Informáticos**

### **2.2.1 Sistema de información gerencial**

Un sistema de información de gestión (MIS) es un sistema informático que consta de hardware y software mismo que sirve como una columna vertebral de las operaciones de una empresa (Guo, 2018). En este sentido, un MIS recopila información de varios sistemas en línea, examina la información y reporta datos para contribuir a la toma de decisiones de gestión (Tabares et al., 2021).

### **2.2.2 Sistema transaccional**

Sistema que permite recuperar, almacenar y modificar la información de una organización para la cual fue diseñada y ser recuperada eventualmente según lo requiera, estas deben mantener la transparencia y seguridad en los datos (Gómez, 2022). Asimismo, Katona & Kovari (2018) mencionan que son sistemas que acceden enmendar errores existentes, accediendo dar marcha atrás en las operaciones realizadas.

### **2.2.3 Sistema de apoyo a las tomas de decisiones**

Un sistema de soporte de decisiones (DSS) es un programa computarizado que se utiliza para respaldar determinaciones, juicios y cursos de acción en una organización o negocio. Un DSS analiza y filtra grandes volúmenes de información, recopilando datos que pueden utilizarse para la resolución de problemas y poder tomar buenas decisiones (Souza et al., 2021).

No obstante, Psarommatis & Kiritsis (2022) manifiestan que los DSS emplean información de manera resumida, las tendencias, las excepciones y los patrones manejando modelos analíticos. Por lo tanto, para la toma de decisiones se debe recolectar información importante, por medio de datos en bruto, conocimiento de modelo de negocios y documentación existente.

## **2.2.4 Estratégicos**

Conocidos también como MRP Manufacturing Resoure Planning, son sistemas cuya principal función es contraria a las anteriores, las mismas que son desarrolladas dentro de casa, estas se acomodan a paquetes existentes en el mercado, esto se debe a que se trata de una parte importante en este tipo de sistemas, mientras a estos sistemas se les aumente funciones o procesos consienten el desarrollo de la empresa (Samantha, 2018, pág. 27).

## **2.3 Java**

Aunque se usa principalmente para aplicaciones basadas en Internet, Java es un lenguaje simple, eficiente y de propósito general. Java se diseñó originalmente para aplicaciones de red integradas que se ejecutan en múltiples plataformas (Yuan & Banzhaf, 2018). Es un lenguaje portátil, orientado a objetos e interpretado. Puede ejecutar con seguridad un subprograma de Java descargado de Internet, porque las funciones de seguridad de Java evitan que este tipo de subprogramas accedan al disco duro de una PC o las conexiones de red. Un subprograma suele ser un pequeño programa Java que está incrustado en una página HTML (Beyer, 2019).

Las aplicaciones Java generalmente se compilan en un código de bytes que se puede ejecutar en cualquier máquina virtual Java (JVM), independientemente de la arquitectura informática subyacente. La sintaxis de Java es similar a la de C y C++, pero tiene menos funciones de bajo nivel que cualquiera de ellos (Brown, & Altadmri, 2017). Con ello, la ejecución en Java brinda capacidades dinámicas que habitualmente no se encuentran en lenguajes de programación tradicionales (Lovato, 2018).

### **2.3.1 Historia De Java**

Según Brown, & Altadmri (2017) mencionan que Java fue desarrollado por Sun Microsystems a principios de la década de 1990. Posteriormente se convirtió en Oracle a través de la compra de los derechos, creo la primera versión de Java que permitía el control de electrodomésticos, PDAs y pequeños ordenadores, esto fue en el año de 1990 su creador James Gosling, que en sus inicios lo nombre Oak. En el año de 1992 el proyecto obtuvo el nombre Star 7, mismo que presentaba un elevado costo en su desarrollo, esto contribuyó a que Sun Microsystem piense en una disolución. Gracias a Naughton, genera oportunidades, y gracias a la creación de un sistema operativo base, y a

la organización que decidió el cese del proyecto y a la aparición de la WEB, Naughton inicia con el desarrollo de los que hoy se conoce como Java (Beyer, 2019). En el año de 1994 se termina con el prototipo nombrado Hot Java que desde su apertura se lo reconoce como un lenguaje de programación potente, por soportar una multiplataforma y que el código era soportado por la WWW, a inicios de 1995 Gage y Andreessen terminan con una versión Java denominada alpha que hasta finales del año anterior solo era soportado por Solaris, y con esta versión podía ser usado en la WWW desde cualquier navegador (Yuan & Banzhaf, 2018). Actualmente, se puede encontrar dicha tecnología en múltiples dispositivos, teléfonos, superordenadores, portátiles y una diversidad de más aplicaciones (Samantha, 2018).

### **2.3.2 Api y Servicios**

Java EE se encuentra construido sobre la base de Java SE, incorporando la funcionalidad y proporcionando al desarrollador los servicios necesarios para crear una aplicación de organización que presenta excelentes características de portabilidad, escalabilidad e integración con tecnologías anteriores (Edwards et al., 2017). Conjuntamente, el servidor de aplicaciones admite manejar transacciones, ocurrencia, seguridad y gestión de los componentes utilizados, y de esta forma el desarrollador logra ocuparse más en la lógica de negocio de los componentes que en tareas de mantenimiento de más bajo nivel (Savitch, 2019). En base a la documentación oficial de Oracle los servicios más usuales que se proporcionan y las APIs asociadas a dichos servicios se describen a continuación:

- EJBs (javax.ejb): definen las clases e interfaces necesarias para manejar los Enterprise JavaBeans (EJBs). Se detallan las interacciones entre los EJBs y sus clientes, y entre los EJBs y el contenedor. Los EJBs suministran servicios de concurrencia, comunicación, transacciones y control de acceso.
- JAXP (javax.xml): los paquetes establecen el API de Java for XML Processing.
- JCA (javax.resource): este paquete define la Java EE Connector Architecture (JCA). Esta tecnología permite enlazar las aplicaciones en los servidores con los sistemas de información internos de las organizaciones.
- JDBC (java.sql y javax.sql): Los paquetes java.sql y javax.sql definen el API de Java DataBase Connectivity (JDBC). Es un API que manipula la conectividad de los programas Java con las bases.

- JMS (javax.jms): el API de Java Messaging Service (JMS) suministra una manera común para que los programas Java creen, envíen, reciban y lean los mensajes de un sistema de comunicación empresarial.
- JNDI (javax.naming): definen la API de la Interfaz de Nombrado y Directorio Java. Admite a los clientes buscar y descubrir componentes y recursos a través de un nombre lógico, independientemente de la aplicación subyacente.
- JPA (javax.persistence): proporciona las clases e interfaces que modelan la interacción entre proveedores de persistencia, clases administradas y clientes del Java Persistence API (JPA).
- JSF (javax.faces): Java Server Faces suministra un entorno para facilitar el desarrollar las interfaces de usuarios, empleando aplicaciones Java.
- JTA (java.transaction): estos paquetes definen la Java Transaction API (JTA). JTA contempla varias interfaces de Java que abarca las partes involucradas en el sistema de transacciones distribuidas y el manejador de transacciones.
- JWS (javax.jws): el Java Web Service (JWS) suministra los elementos e interfaces necesarios para que las aplicaciones interactúen con Servicios Web.
- Security Service (javax.security): se abordan temas de seguridad como el control de acceso a recursos autenticación, protegidos y autorización, y otros.
- Servlets y JSPs (javax.servlet): define un conjunto de APIs usadas principalmente para tratar peticiones HTTP. En el paquete se contiene también la especificación JSP para aspectos más de presentación (Delgado et al., 2017).

### 2.3.3 Applets

Es un programa que se puede incrustar en cualquier documento en formato HTML. Son ejecutadas en un contenedor, que es proporcionado por un plugin, aplicaciones móviles u otros, que soporten el modelo de programación mediante applets (Savitch, 2019).

### 2.3.4 Servlets

Un servlet es una clase de lenguaje de programación Java que se utiliza para ampliar las capacidades de los servidores que albergan aplicaciones a las que se accede mediante un modelo de programación de solicitud-respuesta (Savitch, 2019). Los servlets continuamente responden a cualquier solicitud, que habitualmente se emplean en el

ampliamiento de aplicaciones que se encuentran alojadas en servidores web. De esta forma, Java Servlet define clases de servlet específicas de HTTP para tales aplicaciones (Samantha, 2018, pág. 31).

Se utilizan para manejar la solicitud obtenida del servidor web, procesar la solicitud, producir la respuesta y luego enviar una respuesta al servidor web. Las propiedades de los servlets son las siguientes: funcionan del lado del servidor y son capaces de manejar solicitudes complejas obtenidas del servidor web (Edwards et al., 2017).

## **2.4 Servidor De Aplicaciones Glassfish**

GlassFish o comúnmente denominado Pez de Cristal, nombre seleccionado por la transparencia que los fundadores le dieron al proyecto, el cual aplica una licencia Open Source denominada: licencia la GNU Public License (GPL) v2 y la Common Development and Distribution License (CDDL) v1.0 (Santos et al., 2019).

### **2.4.1 GlassFish**

GlassFish se trata de un software gratuito, adquirido por Oracle en el año de 2010. Además, utiliza una licencia a modo dual: la Licencia Común de Desarrollo y Distribución (CDDL) y la Licencia Pública General GNU (GPL). Este proyecto ayuda a los desarrolladores a crear tecnologías escalables y convenientes, así como también de servicios adicionales que pueden ser instalados según sea la necesidad (López, 2021).

Asimismo, GlassFish proporciona confiabilidad y rendimiento de nivel empresarial con agrupamiento completo y tiene una amplia gama de funciones (Rumyancev et al., 2021). Es genial porque permite administrar el repositorio central a través del panel de administración, que admite todas las funciones disponibles en GlassFish. Sin embargo, como se sabe, más funciones = más recursos (Delgado, Balbuca, & Ortiz, 2017)

## **2.5 Programación**

Java admite crear de primera clase los objetos, el mismo que tiene varias cualidades que diferencia a Java de otros lenguajes según Abd et al. (2021):

- Ayuda a crear código sólido.
- Gracias a que no es un lenguaje puro para POO, el programador debe ejecutar alguna. disciplina de programación.
- Permite exponer el funcionamiento del código de un objeto o de cualquier otro.
- Permite la mezcla de objetos con los tipos primitivos.
- Proporciona herramientas necesarias para la POO.

### 2.5.1 Palabras reservadas

Este lenguaje de programación presenta una gran cantidad de palabras reservadas como son:

abstracto	doble	conmutador
afirmar	implementa importa	else
booleano	instancia de	enumer
interrupción	int interfaz	inválido
byte caso	largo nativo	extiende
capturar	nuevo	final
caract. clase	paquete	proteido
const.	privado	públio
	ir a	retorno
continuar	si	corto
predeterminado hacer		estático
		strich
		súper

### 2.5.2 Comentarios en el código

En Java los comentarios pueden ser de una línea que se extiende y solo se necesita colocar // o múltiples los que inician con /\* y cierran con \*/ (Samantha, 2018, pág. 32).

### 2.5.3 Empaquetado de objetos

Cuando de aplicaciones complejas tratamos el construir estas mediante paquetes es la manera más fácil de hacerlo. Java se llama colisión de nombres esto se refiere a

expresar dos conceptos diferentes con un mismo nombre. Las áreas que se encapsulan se definen con nombres únicos, pero cuando salen del área ya no lo son (Martínez et al., 2021).

#### **2.5.4 Definición de paquete**

Un paquete es un espacio de nombres que organiza un conjunto de clases e interfaces relacionadas. Conceptualmente, puede pensar que los paquetes son similares a diferentes carpetas en su computadora (Martínez et al., 2021). Puede mantener páginas HTML en una carpeta, imágenes en otra y scripts o aplicaciones en otra (Brown & Altadmri, 2017). Ya que el software Java puede componerse de múltiples clases individuales, es importante tener todos sus componentes organizados, de esta forma se debe colocar las interfaces y las clases relacionadas en los paquetes que sean necesarios (Yuan & Banzhaf, 2018).

De esta manera, la palabra clave para definir un paquete seguida por el nombre y termina en punto y coma. Estos son separados con un punto y seguidos por un plan de facto:

```
package orgType.orgName.appName.compName;
```

Esta definición de paquete se divide de la siguiente manera:

package palabra reservada.

orgType es el tipo de organización, tales como com, org o net.

orgName es el nombre del ámbito de la organización, tales como makotogroup, sun o ibm.

appName es la abreviación de la denominación de la aplicación.

compName es el nombre del componente.

Pero no es necesario seguir estos grupos de paquetes, especificar el paquete no es necesario, esto no requiere que todos los paquetes deben tener nombres de clases únicos y residirán en el paquete predeterminado (Martínez et al., 2021).

### 2.5.5 Sentencias de importación

Para definir un objeto considerando el ejemplo anterior, y se definirá como sentencia de importancia el cual comunica al compilador, el cual se encuentra las clases que se hacen referencia en el código, así se puede precisar objetos no triviales, mismos que utilizan objetos que admiten su funcionamiento (Psarommatis & Kiritsis, 2022).

Una sentencia de importación normalmente de la siguiente manera:

```
1 import ClassNameToImport;
```

### 2.5.6 Declaración de clase

Los objetos pueden ser declarados en una clase, se utiliza una plantilla por cada uno de los objetos, en la clase se define una estructura básica del objeto, su aplicación crea una instancia del objeto (Beyer, 2019).

```
accessSpecifier class ClassName {  
  
    accessSpecifier    dataType    variableName [=    initialValue];  
        accessSpecifier  
    ClassName([argumentList]) { constructorStatement(s)  
    }  
    accessSpecifier returnType methodName([argumentList]) { methodStatement(s)  
    }  
}
```

Un accessSpecifier de una clase el cual podíamos tener tantos valores que en lo general, es público.



### 2.5.7 Variables

Estas variables son empleadas solamente en casos necesarios. En Java no está restringido su utilización, en estos casos es el programador el que debe definir dónde y cuándo usarlos:

- Privado: pueden ser visualizadas solo por la clase que contiene la variable.
- Protegido: pueden ser visualizadas solo por aquellos objetos que tengan autorización de usarlos.
- Público: pueden ser visualizadas por cualquier objeto.

### 2.5.8 Métodos

En Java el comportamiento de la clase está definido por la misma. En algunos instantes solamente devuelve los valores del atributo, sin embargo, puede ser complejo. De esta manera, existen dos clases de métodos: los constructores que permiten instanciar la clase y los destructores que permiten la liberación de las mismas (Edwards et al., 2017).

### 2.5.9 Sintaxis de la declaración del constructor.

```
accessSpecifier ClassName([argumentList]) {  
    constructorStatement(s)  
  
    }
```

### 2.5.10 Métodos estáticos y de instancia

Existen también otros métodos como son de instancia y estáticos. En este sentido, los de instancia son los que dependen del estado del objeto mediante su comportamiento, por otra parte, los estáticos se usan en gran medida dependiendo de los objetos (Savitch, 2019).

### 2.5.11 Adición de variables de clases

Dentro de una clase pueden estar definidas múltiples variables, mismas que contiene un accessSpecifier, un dataType datos primitivos que referencian a un objeto, un variableName, de manera opcional un initialValue (Psarommatis, & Kiritsis, 2022).

### 2.5.12 Tipos de datos primitivos

Tipo	Tamaño	Valor predeterminado	Rango de valores
booleano	n/d	falso	verdadero o falso
byte	8 bits	0	-128 a 127
caract.	16 bits	(sin firmar)	\u0000' \u0000' a \uffff' o 0 a 65535
corto	16 bits	0	-32768 a 32767
int	32 bits	0	-2147483648 a 2147483647
largo	64 bits	0	-9223372036854775808 a 9223372036854775807
flotante	32 bits	0,0	1.17549435e-38 a 3.4028235e+38
doble	64 bits	0,0	4.9e-324 a 1.7976931348623157e+308

**Tabla 1** Tipos de datos primitivos de Java

**Fuente:** (Samantha, 2018)

### 2.5.13 Cadenas y operadores

A continuación, se explica sobre las cadenas y cuándo y cómo usar los operadores.

### 2.5.14 Cadenas

Según Gómez (2022) menciona que otros tipos de variables son de tipo texto las mismas que son manejadas como cadenas en forma de matrices. Una cadena se denomina una secuencia de caracteres. El paquete java.lang viene incorporado en Java, mismo que

abarca las estructuras de información de una determinada cadena, es decir Java presenta múltiples características que lo componen para manejar las cadenas existentes.

### 2.5.15 Operadores

En Java los operadores aritméticos permiten ejecutar las diferentes operaciones matemáticas, como la resta, suma, división, multiplicación, entre otras (Tabares et al., 2021). Como cualquier otro lenguaje de programación Java presenta operadores que admiten ejecutar un sin número de operaciones como:

Operador	Uso	Descripción
+	$a + b$	Suma a y b
+	$+a$	Potencia a int si es un byte, short, o char
-	$a - b$	Resta b de a
-	$-a$	Aritméticamente niega a
*	$a * b$	Multiplica a y b
/	$a / b$	Divide a por b
%	$a \% b$	Devuelve el resto de la división de a por b (el operador de módulo)
++	$a++$	Incrementa a por 1; calcula el valor de a antes de incrementarlo
++	$++a$	Incrementa a por 1; calcula el valor de a después de incrementarlo
--	$a--$	Disminuye a por 1; calcula el valor de a antes de disminuirlo
--	$--a$	Disminuye a por 1; calcula el valor de a después de disminuirlo

Operador	Uso	Descripción
+=	a += b	Taquigrafía para a = a + b
-=	a -= b	Taquigrafía para a = a - b
*=	a *= b	Taquigrafía para a = a * b
%=	a %= b	Taquigrafía para a = a % b

**Tabla 2** Operadores lógicos

**Fuente:** (Samantha, 2018)

Operador	Uso	Retorna verdadero si
>	a > b	a es mayor que b
>=	a >= b	a es mayor que o igual a b
<	a < b	a es menor que b
<=	a <= b	a es menor que o igual a b
==	a == b	a es igual a b
!=	a != b	a no es igual a b
&&	a && b	Ambos a y b son verdaderos, evalúa condicionalmente a b (si a es falso, b no se evalúa)
	a    b	a o b es verdadero, evalúa condicionalmente a o b (si a es verdadero, b no se evalúa)
!	!a	a es falso

**Tabla 3** Operadores aritméticos, relacionales y condicionales del lenguaje Java

**Fuente:** (Samantha, 2018)

## 2.5.16 Sentencia de control

### IF

Admite realizar operaciones de pregunta si sintaxis:

```
if (gender.equals("MALE"))
```

### Else

Cuando la pregunta controlada por el IF es de tipo negativa permite dar otra opción de resultado.

```
if
```

```
    (gender.equals("MALE"  
    )) ret = height +  
2; else {  
  
    ret = height;  
  
    Logger.getLogger("Person").info("Being honest about height...");
```

### Bucles

Cuando en la programación se necesita que repetidamente se cumpla con algunas condiciones se utilizan los bucles:

### for

Permite controlar en un rango cuantas veces se ejecute el mismo

```
for (initialization; loopWhileTrue;  
    executeAtBottomOfEachLoop) { statementsToExecute  
  
    }
```

### while

La sintaxis para un bucle while

```
es: while (loopWhileTrue) {  
statementsToExecute  
}
```

Se ejecuta mediante el resultado sea verdadero, mientras sea verdadero el bucle continuara.

### **do...while**

si se quiere un bucle que se ejecute siempre una vez y luego verifique su expresión condicional, se puede utilizar un bucle do...while (Steven, 2012).

```
do {  
  
    Person p = new Person("Joe Q Author", 42, 173, 82,  
        "Brown", "MALE"); l.info("Loop executing iteration# " +  
        aa); l.info("Name: " + p.getName());  
  
    l.info("Age:" + p.getAge());  
  
    l.info("Height (cm):" + p.getHeight());  
  
    l.info("Weight (kg):" +  
  
        p.getWeight()); l.info("Eye Color:" +  
        p.getEyeColor()); l.info("Gender:"  
  
        + p.getGender()); aa++;  
  
} while (aa < 3);
```

## **2.6 Otros Lenguajes De Programación**

### **2.6.1 Lenguaje simple**

En el mercado existen múltiples lenguajes de programación, sin embargo, lo que hace especial a Java es su simplicidad, es decir que su curva de aprendizaje es corta, esto se logra porque los programadores pueden aprender y familiarizarse con mayor rapidez de las funciones y términos del lenguaje (Cruttwell et al., 2021).

### **2.6.2 Lenguaje orientado a objetos**

Es importante considerar que no todos los lenguajes son POO, característica que convierte a JAVA en un lenguaje de programación 100% POO, esto se debe a que los

objetos se encapsulan en clases y funciones, mismos que pueden ser manipulados, convirtiéndose en mucho más potentes (Saleh et al., 2021).

### 2.6.3 Aplicaciones distribuidas

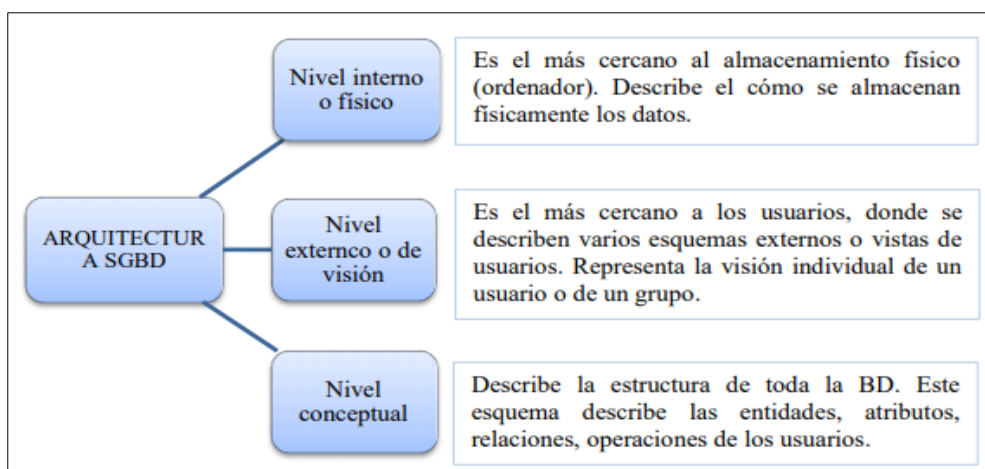
Cruttwell et al. (2021) mencionan que con el lenguaje de programación de Java se puede efectuar aplicaciones distribuidas en la red, mismas que se ejecutan en la plataforma que tiene como base un componente de cómputo distribuido y asimismo mantiene el rendimiento y la estabilidad considerable de la aplicación.

### 2.6.4 Interpretado y compilado

El lenguaje de programación de Java presenta una gran ventaja sobre los demás, su compilación es excelente, igualándose a lo que ocurre en el lenguaje ensamblador. Es decir, la interpretación desde la base, ayuda muchísimo en la ejecución de las distintas aplicaciones (Singh et al., 2021).

## 2.7 Arquitectura

“El comité ANSI-SPARC propuso una arquitectura de tres niveles para los SGBD”. Misma que se ha convertido en un estándar para los SGBD al ser la más empleada. Con ello, la arquitectura presenta como objetivo separar los programas de aplicaciones de bases de información de forma física (Shan et al., 2022). El esquema de una base de datos se define en tres niveles de abstracción distintos descritos a continuación:



**Figura 2.1** Arquitectura de los SGBD

**Fuente:** Víctor Colcha

### 2.7.1 Componentes

El sistema de base de datos está dividido por módulos, mismos que controlan cada una de las partes de un sistema (Chen et al., 2021). De esta manera, entre los componentes funcionales de un sistema de base de datos se encuentran:

- **Compilador de DDL:** transforma las proposiciones DDL en un compilados de tablas que disponen de metadatos.
- **Diccionario de datos:** presenta información relacionada con la estructura de la base de datos.
- **Gestor de archivos:** gestiona la asignación de espacio en las estructuras de datos usadas para representar datos y en la memoria del disco.
- **Índices:** permiten crear accesos rápidos a registros que contemplan valores específicos.
- **Administrador de base de datos:** utilizado de interfaz entre los programas de aplicación y los datos.
- **Procesador de consultas:** explica las proposiciones existentes en el lenguaje de consulta de manera fácil (Colcha, 2015, pág. 11).

### 2.7.2 PostgreSQL

Es un sistema de base de información de tipo relacional de objetos que utiliza código abierto. Presenta una arquitectura comprobada para lograr una reputación de corrección, confiabilidad e integridad de datos. Asimismo, PostgreSQL fue desarrollado por un equipo de desarrolladores voluntarios a nivel mundial, y no se encuentra controlado por ninguna empresa privada, siendo su código fuente gratuito (Woltmann et al., 2021). De tal forma, Vilorio et al. (2019) indican que PostgreSQL se ejecuta en todos los principales sistemas operativos, incluidos Linux, Solaris, Windows, entre otros. Asimismo, soporta sonidos, texto, videos e imágenes, incluyendo interfaces de programación para Open Database Connectivity (ODBC), Java, C/C++, Ruby, Tcl y Python.

### 2.7.3 Características

Las características soportadas por el SGBD PostgreSQL se encuentra divididas por niveles:

**Nivel general:**



- Admite el acceso encriptado vía SSL, licencia BSD, grande y diversa documentación.
- Es una base de datos 100% ACID y controla la integridad referencial
- Posee Tablespaces y Nested transactions (savepoints), replicación asincrónica/sincrónica y efectúa copias de seguridad en caliente (Online/hot backups).
- Posee Juegos de caracteres internacionales, Unicode y Regionalización por columna.
- Utilizable para UNIX y Linux en todas sus versiones.

#### **Nivel programación:**

- Admite crear funciones/procedimientos almacenados en numerosos lenguajes de programación, entre otros PL/pgSQL (similar al PL/SQL de Oracle), PL/Perl, PL/Python y PL/Tcl
- Contiene APIs para programar en Tcl, C/C++, Java, ODBC, Python, .Net, Perl, Ruby, Scheme, PHP, Qt, Lisp, entre otros.
- Contiene varios tipos de datos que posibilitan definir otros tipos. Entre los tipos estándares de información se encuentran matrices, cadenas binarias, etc.
- Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido).

#### **2.7.4 Ventajas**

- Extensible: el código fuente se encuentra disponible de manera gratuita, para todo aquel que necesite personalizar o extender lo pueda realizar sin ningún costo.
  - Instalación ilimitada: no se puede demandar a una organización por ser instalado en más ordenadores de los que la licencia permite, puesto que no hay costo asociado a la licencia de software.
  - La respuesta en bases de datos pequeñas puede ser deficiente.
  - Los ahorros de costo son considerables: fue diseñado para poseer un mantenimiento eficiente y de ajustes menores que los productos de distribuidores comerciales.
  - Multiplataforma: se encuentra disponible en casi todos los sistemas operativos.
- Desventajas**
- El nivel de consumo de los recursos es más que de MySQL.

- Es más lento en actualizaciones e inserciones relación con MySQL.
- En cuanto a la sintaxis en algunas de los comando o sentencias no son nada intuitivos.
- Soporte en línea: existen foros oficiales, sin embargo, no existe una ayuda obligatoria.

### **2.7.5 PgAdmin**

Es la plataforma de desarrollo y administración de código abierto más popular y rica en funciones para PostgreSQL, la base de datos de código abierto más avanzada del mundo. Esta puede ser utilizada en Windows, Linux, Unix y macOS. El software está disponible en formato fuente y binario desde la red espejo de PostgreSQL.(Freddy, 2020).

### **2.7.6 NetBeans**

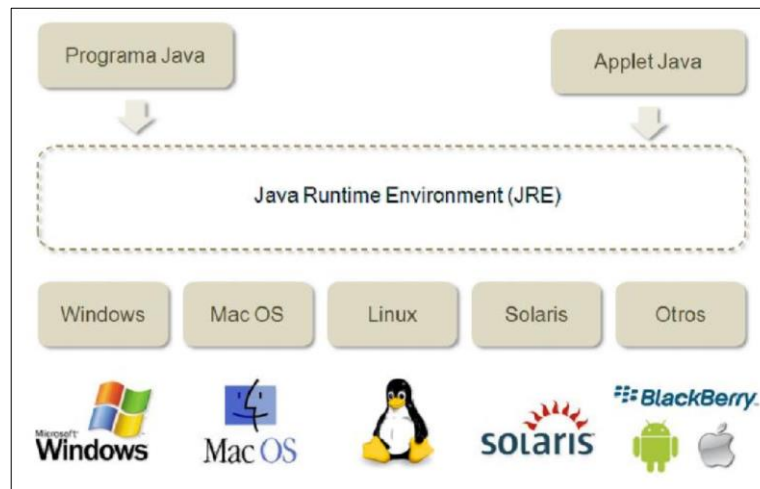
#### **2.7.6.1 Introducción**

NetBeans IDE es un entorno de desarrollo integrado (IDE) gratuito y de código abierto que le permite desarrollar aplicaciones de escritorio, móviles y web. NetBeans permite el desarrollo de las aplicaciones utilizando múltiples lenguajes de programación que incluyen a Java, C++, PHP y HTML5 (Arzoo et al., 2017). Además, contiene soporte integrado para cubrir todo el desarrollo, desde el inicio del proyecto, la depuración, la creación de los perfiles e implementación del sistema. Es importante mencionar, que NetBeans se ejecuta por medio de Linux, Windows, Mac, entre otros sistemas (Dhika et al., 2019).

Wijaya (2019) menciona que NetBeans permite dar un soporte de manera integral para tecnologías JDK 7 y las actualizaciones de Java dadas recientemente. Es completamente compatible con Java EE empleando los estándares actuales para Java, servicios web, XML, SQL, GlassFish Server.

NetBeans administra las siguientes componentes y características existentes en la plataforma: almacenamiento, ventanas, asistente de marco, ajustes de usuario, biblioteca visual de NetBeans, herramientas de desarrollo integradas. Asimismo, usa componentes denominados módulos, que ayudan al desarrollo de software (Arzoo et al., 2017).

Su instalación es compatible para diferentes sistemas operativos: Windows, Linux, Mac OS, etc. como se muestra en el siguiente gráfico:



**Gráfico 2** Entorno desarrollador abierto para algunos sistemas operativos

**Fuente:** Netbeans Java

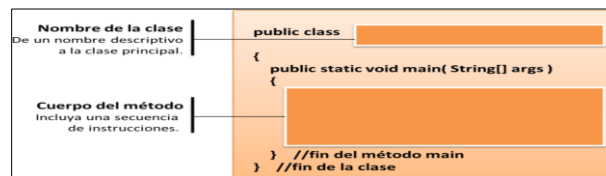
### 2.7.6.2 Desarrollador Java NetBeans

Java se caracteriza por estar orientado a objetos en su lenguaje de programación proponiendo su independencia de implementos como fuera posible. Su objetivo es que los desarrolladores escriban una programación una sola vez y pueda ser ejecutados en cualquier dispositivo existente. Resulta necesario para el uso de Java ser desarrollado de NetBeans las clases JDK, junto con sus complementos de librerías y bibliotecas estándar (Ardian et al., 2018).

### 2.7.6.3 Estructura de la programación Java

```
public class ClasePrincipal {public static void main (String [] args) {  
    sentencia_1;  
        sentencia_2;  
        // ...  
        sentencia_N;  
    }  
}
```

#### 2.7.6.4 Programación detallada en Java



**Gráfico 3** Estructura básica detallada en java

**Fuente:** NetBeans Java

**Definición de clase:** la primera línea del programa, luego del primer comentario. Define una clase, la definición de la clase empieza por el carácter {y acaba con el carácter}. El programador define el nombre de la clase (Soares & Souza, 2017).

**Definición del método:** posterior a la definición de clase se describe la definición del método `main ()`. Todos los programas Java necesariamente incluyen un método `main ()`. Este señala las sentencias a efectuar cuando se ejecuta un programa. En este sentido, las sentencias de un método están delimitadas por los caracteres {y} que indican el inicio y el fin del método, respectivamente (Yuan & Banzhaf, 2018).

**Sentencia:** en el método `main ()` este contiene una sentencia para colocar un texto por la consola. En Java las sentencias de un programa concluyen con el símbolo punto y coma, que indica que ya ha finalizado una sentencia (Llanga, Haro, & Inca, 2016).

#### 2.7.6.5 Ventajas

Asiste parcialmente en la escritura del código de programación.

- Ayuda en la navegación de las clases que son preestablecidas en la plataforma.
- El trabajo es a nivel de archivo, y no solo al nivel del proyecto.
- Los módulos son posibles de ser creados de forma independiente.
- NetBeans es compatible con C, PHP, Java.
- Siendo costoso el aprendizaje, las dificultades son superadas por los beneficios.
- Suprime algunas de las tareas que en proyectos de gran magnitud son pesadas (Savitch, 2019).

#### 2.7.6.6 Que contiene un proyecto en NetBeans

- Archivos con el código.

- Bibliotecas externas (p.e. ACM TaskForce) Imágenes, sonidos, etc.
- Físicamente una organización en particular denominada directorio (Eck, 2021).

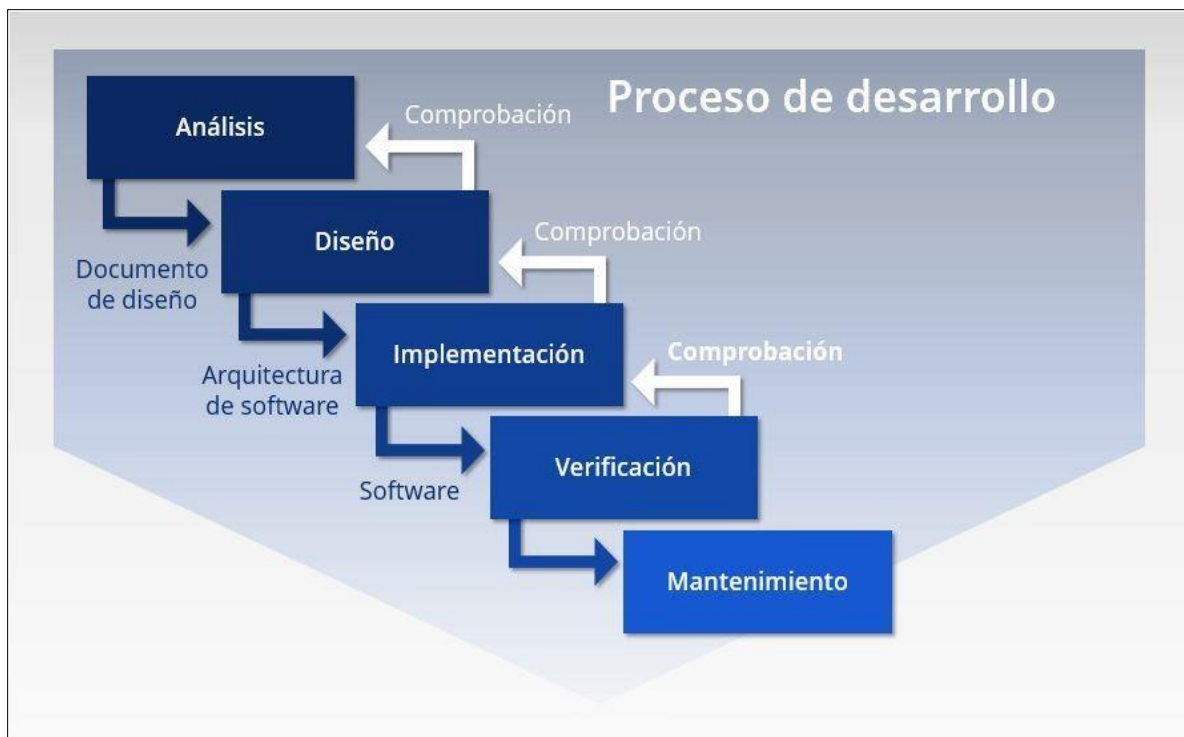
#### **2.7.6.7 Compatibilidad con Java y MySQL**

- MySQL es compatible NetBeans, al integrar una librería que admite la conexión.
- Presenta asistencia para configuración y creación de diversos proyectos.
- Contiene una herramienta que limpia errores en el código, esto es por el debugger que incluye el IDE.
- Con el editor de código ayuda a comprobar la sintaxis y semánticas para la creación de aplicaciones exitosas.
- Las vistas y ayudas que estructura la visualización contribuyen a mostrar en Java diversas pantallas con código, con diversas vistas con las clases y opciones diversas.
- NetBeans da soporte a las novedades de Java (Samantha, 2018).

**CAPÍTULO III**  
**ANÁLISIS Y DISEÑO DEL SISTEMA**

### 3.1 RECOPIACIÓN DE INFORMACIÓN

Para el desarrollo e implementación del sistema informático para el registro de mantenimiento de los equipos de cómputo de GADM del cantón Alausí desarrollado en Java con base de datos PostgreSQL, se empleó la Metodología en Cascada, misma que permitió cumplir con las etapas establecidas y corregir los errores para generar un sistema de calidad.



**Gráfico 4** Metodología de desarrollo

**Fuente:** (Digital Guidi, 2020)

En GADM del cantón Alausí y la necesidad de tener un control óptimo del mantenimiento de los equipos informáticos llevo a la necesidad de automatizar este proceso mediante un sistema informático de registros de mantenimiento de sus equipos informáticos. Por lo tanto, para levantar la información se utilizó una entrevista dirigida al encargado del Departamento de mantenimiento ubicada en la Unidad de TICs del GADM de Alausí, permitiendo la recopilación de información requerida para establecer los requerimientos necesarios en la implementación del sistema. Este sistema ayudará al encargado a cubrir las necesidades del departamento.

De esta manera, el sistema cuneta con varios módulos como son:

- Ingreso de usuarios para el uso del sistema
- Ingreso de equipos para mantenimiento y responsables
- Modificación y actualización de equipos con de datos errados
- Eliminación de equipos registrados
- Generación de reportes por equipos y por usuarios

## **ALCANCE**

El sistema que actualmente cumple con el control de mantenimiento de equipos del GADM Alausí, no permite un control preciso de los equipos al ser un control totalmente manual. En este contexto, se presenta el requerimiento de crear un sistema web que ayude a la automatización del control de los mantenimientos a equipos informáticos, para así dar solución a los varios problemas que han sido dados de baja o que se han extraviado por no contar con el soporte necesario de los mismos.

### **3.2 ANÁLISIS**

En esta fase se efectúa el análisis del sistema para el control del mantenimiento de los equipos de cómputo. El análisis contribuyó a la creación de un sistema que funcione de manera automatizada, con la finalidad de controlar el mantenimiento de los equipos del GADM Alausí, mismo que trata de solucionar las necesidades del departamento de mantenimiento.

Entre las funciones que conforman el sistema, se encuentran:

- Ingreso al sistema mediante una aplicación .exe que muestra la pantalla de ingreso por medio de la cual se digita el usuario y el password previamente ingresados en el sistema.
- Mediante pantallas con los datos controlados que permiten el ingreso de información en varias áreas como: usuarios, mantenimientos y reportes.
- Mediante consultas al BD recuperación y actualización de información puede ser presentada en pantalla o en archivos de impresión.

#### **3.2.1 Estudio de Factibilidad**

Con la finalidad de tomar decisiones en cuanto a la realización del proyecto se elaboró un análisis para determinar con lo que cuenta el GADM de Alausí en relación a:



equipos, personal, software y el equipo de trabajo, todo esto para determinar la viabilidad para la realización del proyecto.

La determinación de la factibilidad estuvo conformada por estudios: operativos, técnicos, económicos y legales.

### 3.2.1.1 Factibilidad operativa

Dentro de la factibilidad operativa se definieron las personas involucradas que requiere el desarrollo e implementación del proyecto informático. En la tabla presentada a continuación se identifica al dueño del producto es decir el usuario final, un desarrollador, un diseñador y un tester.

<b>Rol</b>	<b>Persona</b>	<b>Funciones</b>
<b>El Dueño del producto</b>	Alcalde de Gad Alausí	Aportar los recursos humanos y económicos para el desarrollo del proyecto
<b>Usuario final</b>	Ing. José Luis Sislema	Experto en el proceso que se convierte en la fuente de requisitos
<b>Equipo de Desarrollo</b>	Rene Armando Buñay	Desarrollar el proyecto
<b>Asesor</b>	Ing. Juan Pablo Romero	Asesor experto en el proceso SCRUM
<b>Diseñador</b>	Rene Armando Buñay	Encargado de la maquetación del sitio web

**Tabla 4** Recursos humanos

**Fuente:** (Buñay, 2019)

Una vez analizados los recursos humanos necesarios para el desarrollo e implementación del proyecto se pueden decir que es factible operativamente, sabiendo que el Alcalde de Alausí extendió la autorización para crear el proyecto, el desarrollador por fines académicos realizó el sistema, el asesor es la fuente de conocimiento y guía y el diseñador son proporcionados por el departamento de tecnología del GADM de Alausí.

### 3.2.1.2 Factibilidad técnica

Dentro de la factibilidad técnica se analizaron el software y el hardware necesarios en la creación del sistema informático requerido.

Nombre	Especificación técnica	Incidencia en el desarrollo del sistema
PostgreSQL 11.9	Motor de base de datos	No se incurrió en gasto para su utilización debido a que está desarrollado bajo código abierto, por motivos que es multiplataforma no existió inconvenientes al implementarla. A pesar que el tamaño de la base y la cantidad de datos que maneja es grande, presenta estabilidad y un rendimiento constante en la recuperación de información. Cabe recalcar que para realizar todo el esquema de las bases de datos se utilizó el administrador de la base de datos pgAdmin III, siendo esta una herramienta gráfica y completa, de gran ayuda para la creación de cada uno de los componentes que intervienen en la misma.
NetBeans 8.2	Entorno desarrollo integrado (IDE) de	La utilización de NetBeans en el desarrollo de la aplicación, es por su facilidad de uso y la velocidad con que se puede generar código mediante la combinación de teclas, permitiendo de esta forma tener mayor fluidez en la resolución de cada una de las tareas.
GlassFish 4.0	Servidor web	El servidor de GlasFish se integra de forma correcta y sin complicación alguna con el entorno de desarrollo NetBeans, que a diferencia de otros, se dificulta considerablemente su configuración, además cuenta con una interfaz gráfica para administrar cada una de las aplicaciones a desplegar.

Nombre	Especificación técnica	Incidencia en el desarrollo del sistema
Windows 10	Sistema operativo	El despliegue del sistema se lo realizó mediante la utilización del sistema operativo Windows 10, las configuraciones se las realizaron a través de ventanas, y la ejecución de los programas NetBeans 8.2 y PostgreSQL 11.9
Star Uml 2.7.0	Modelador de diagramas UML	Existen herramienta para el modelamiento de software basado en los estándares UML más potentes y con una amplia variedad en comparación a la seleccionada, pero fue elegida debido que cumple con cada una de las expectativas al momento de realizar cada uno de los modelos requeridos en el proyecto, además de que es un instrumento gratis y su utilización no es de desconocimiento para los desarrolladores.
Mozilla Firefox,	Navegador	Los dos navegadores web señalados pueden ser utilizados debido que no presentan ningún
Google Chrome		inconveniente en interpretar cada uno de los componen que conforman los formularios de la aplicación.

**Tabla 5** Software

**Fuente:** (Buñay, 2019)

Nombre	Especificación Técnica	Estado
Computadora	<ul style="list-style-type: none"> <li>Computadora de Escritorio de la Unidad de UTICs, con Windows 10, que soporte un browser de navegación.</li> </ul>	Aceptable

	<ul style="list-style-type: none"> <li>• Aplicación Web de Escritorio para el almacenamiento de la información y se pueda alojar la base de datos vinculada al sitio web que se encuentra en la máquina servidor de UTICs.</li> </ul>	
--	---	--

**Tabla 6 Hardware**

**Fuente:** (Buñay, 2019)

El GADM de Alausí tiene tanto el software como el hardware necesario para el desarrollo e implementación del sitio web, por lo tanto, según el análisis de la factibilidad técnica es factible realizar el presente proyecto informático.

### 3.2.1.3 Factibilidad económica

En factibilidad económica se detallan los gastos que incurrieron en el desarrollo del sistema:

Cantidad	Detalle	Valor
1	Laptop Hp	600,00
10	Resmas de papel	50,00
3	Tinta de impresión (negro y colores)	60,00
1	Disco Duro externo	90,00
1	Servicios de Internet	100,00
1	Transporte	100,00
1	Revisiones	100,00
3	Empastados	80,00
3	Anillados	20,00
1	Impresora Epson Color	200,00

Otros	Otros gastos	80,00
Total:		1480,00

**Tabla 7** Gastos económicos

**Fuente:** (Buñay, 2019)

Con lo expuesto anteriormente, se puede indicar que para llevar a cabo el proyecto se cuenta con los recursos económicos necesarios por parte de los investigadores, el mismo que determina que es factible el desarrollo del proyecto.

En los costos están los suministros y materiales que se utilizarán dentro del proyecto. En total se obtiene la suma de \$1480,00 como costo del proyecto.

### **3.2.1.4 Factibilidad legal**

El sistema Java con base de datos PostgreSQL es una aplicación de escritorio de uso interno para el GADM Alausí que no con llevan al pago de licencias y que la creación del sistema de mantenimiento será entregada para el uso exclusivo y sin restricciones del GAD del cantón Alausí, con ello evitar problemas legales en un futuro.

### **3.2.2 Análisis de Requerimientos**

Para el desarrollo de la presente investigación se empleó la metodología ágil denominada SCRUM, misma que permitió cumplir con satisfacción el desarrollo del proyecto informático planteado. La metodología requirió que se realice un estudio preliminar, por medio de un levantamiento de información sobre el proceso que se solicita automatizar.

El desarrollo del sistema requiere de varios pasos, enlistados a continuación:

- Especificación de requerimientos
- Instalación de software requerido
- Diseño de la base de datos
- Análisis de los roles del sistema
- Desarrollo de diagramas UML (diagramas de casos de uso, diagrama de secuencia)

- Codificación del sistema
- Pruebas de usuario
- Capacitación del sistema

### **3.2.2.1 Requerimientos funcionales del sistema**

Los requerimientos del sistema informático para la gestión de pedidos y ventas fueron establecidos y priorizados conjuntamente con el Ing. José Luis Sislema, a través de reuniones que permitieron establecer la funcionalidad del sistema, con requerimientos detallados a continuación:

- Autenticar un usuario en el sistema
- Ingresar un usuario
- Modificar usuario
- Eliminar usuario
- Listar usuarios
- Ingresar un equipo de computo
- Modificar un equipo de computo
- Eliminar un equipo de computo
- Cambios de estados para los mantenimientos
- Listar equipos de computo
- Registrar un mantenimiento realizado
- Modificar un mantenimiento realizado
- Eliminar un mantenimiento realizado
- Listar mantenimientos realizados
- Reporte de mantenimientos realizados a un equipo de cómputo específico
- Reporte General de mantenimientos dado un rango de fechas
- Reporte de mantenimientos realizados por un mismo usuario responsable
- Reporte de equipos de cómputo con un número específico de mantenimientos

### **3.2.2.2 Requerimientos no funcionales del sistema**

- Usabilidad
- Funcionalidad
- Disponibilidad

- Seguridad
- Escalabilidad

Se presentaron un total de 21 requerimientos, de los cuales 16 son funcionales y 5 son no funcionales. Estos requerimientos permitieron efectuar las acciones requeridas para la creación del nuevo sistema.

### 3.2.3 Definición de Casos de Uso

#### 3.2.3.1 Actores

Los actores del sistema informático son:

- **Administrador:** tiene el control total del sistema, y será aquel que tenga permisos y registro en el sistema.
- **Invitado:** es un usuario extra que solo tendrá acceso a la visualización de reportes el mismo que debe ser registrado previamente por el Administrador.
- **Usuario:** se define cuando el sistema puede ser usado por el Administrador y el Invitado.

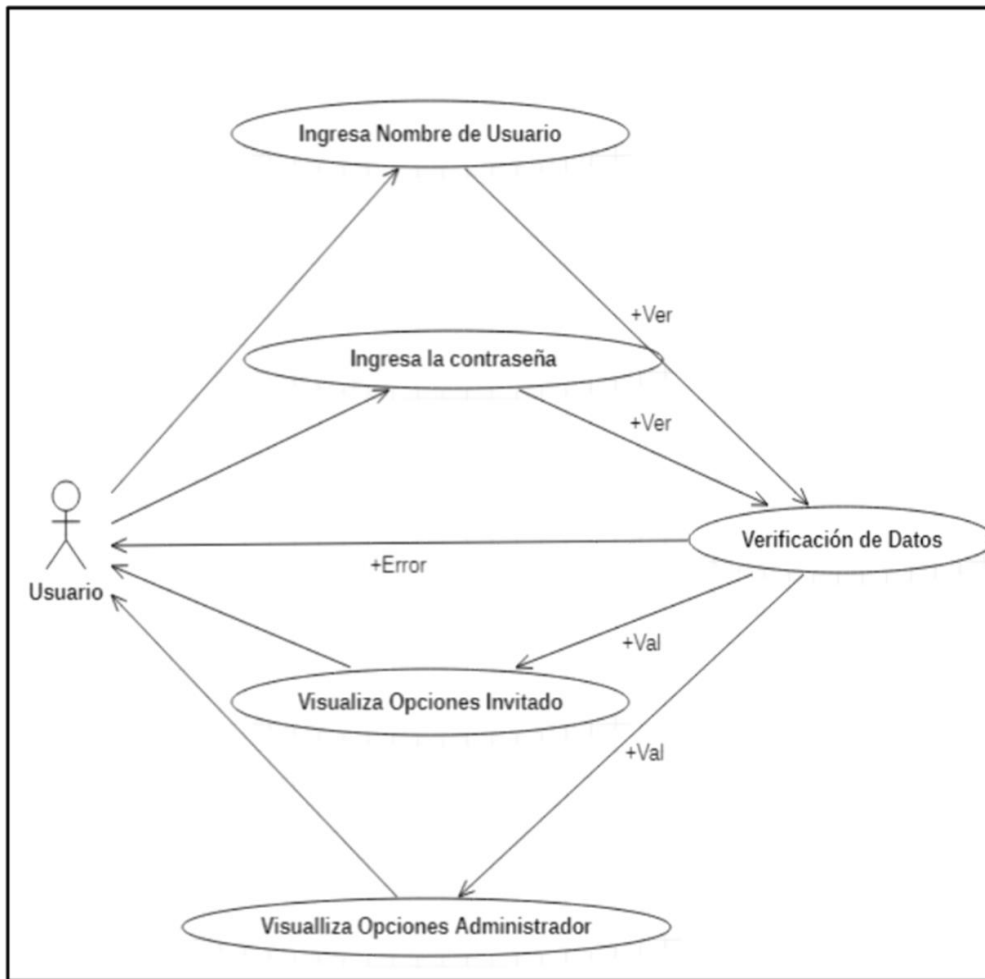
#### 3.2.3.2 Login

**Objetivo:** brindar la apertura para ingresar al sistema desarrollado, dando como resultado la visualización de las opciones para cada usuario.

**Proceso:**

- Ingreso de datos
- Consulta en el BD usuario existente
- Verificación de datos de acceso.

Es importante mencionar que si es Administrador se visualiza opciones completas, por otro lado, si es Invitado solo opción de reportes.



**Gráfico 5.** Ingreso al sistema

**Fuente:** (Buñay, 2019)

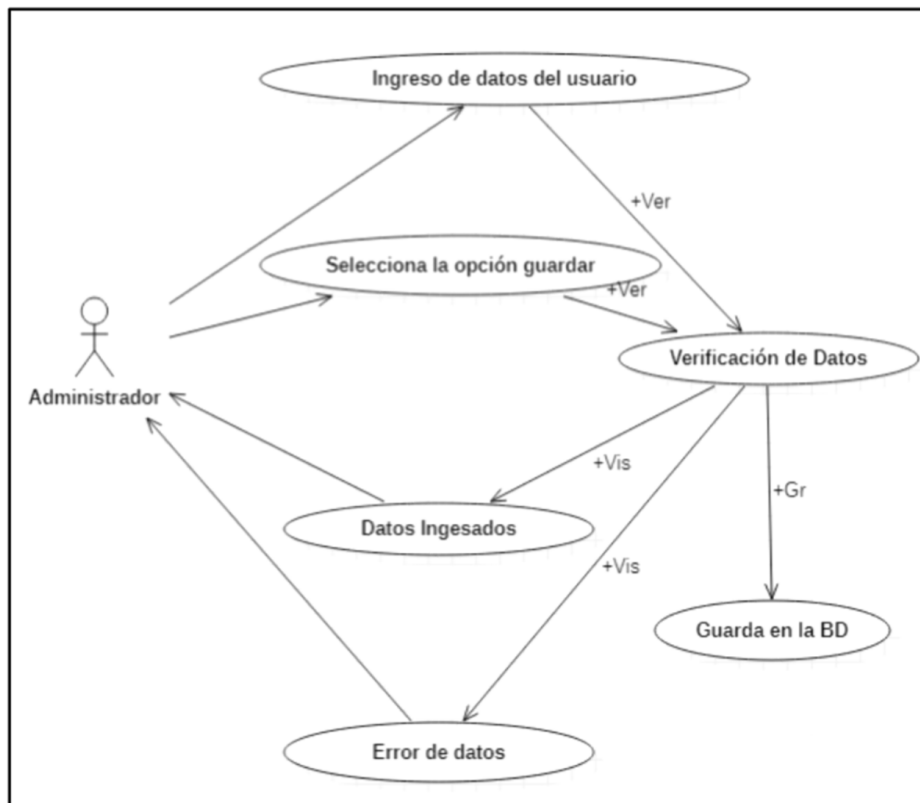
### 3.2.3.3 Registro de datos en el sistema

**Objetivo:** Registrar datos en el sistema y guarda datos en la Base de Datos.

**Proceso:**

- Presentar formulario
- Llenar datos solicitados
- Guarda datos en BD
- Muestra mensaje de almacenamiento
- Se visualiza en pantalla el nuevo registro
- selección opción nueva





**Gráfico 6.** Registro de datos en el sistema

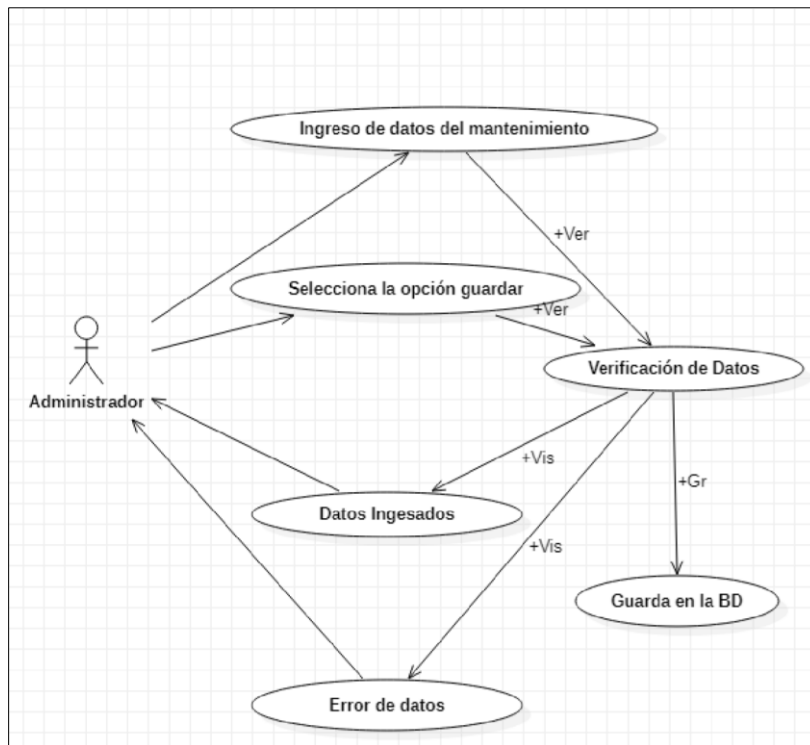
**Fuente:** (Buñay, 2019)

### 3.2.3.4 Registro de mantenimiento

**Objetivo:** permite el registro del mantenimiento de los equipos y así dar un mejor seguimiento a estos mantenimientos.

**Proceso:**

- Ingreso de datos a consultar
- Busca en a BD
- Presenta datos encontrados
- El administrador ingresa datos del mantenimiento
- Guarda los datos
- Entrega la respuesta de datos ingresados correctamente



**Gráfico 7** Registro de mantenimiento

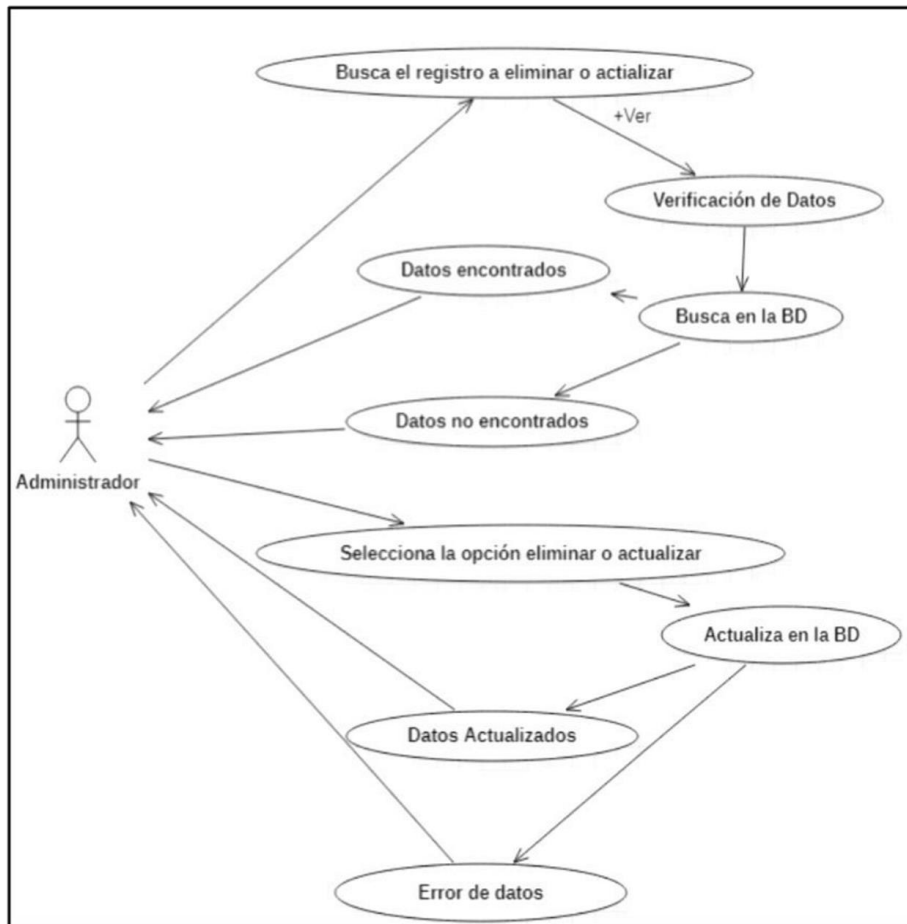
**Fuente:** (Buñay, 2019)

### 3.2.3.5 Eliminación y edición de datos

**Objetivo:** Permite la eliminación y edición o actualización de usuarios, equipos de cómputo y mantenimientos para poder dar un mejor seguimiento a los procesos.

**Proceso:**

- Seleccionar la opción eliminar o editar
- Ingreso de datos a consultar
- Busca en a BD
- Presenta datos encontrados
- Elimina o edita los datos seleccionados
- Guarda los datos
- Actualiza la Base de Datos



**Gráfico 8** Eliminación y edición de datos

**Fuente:** (Buñay, 2019)

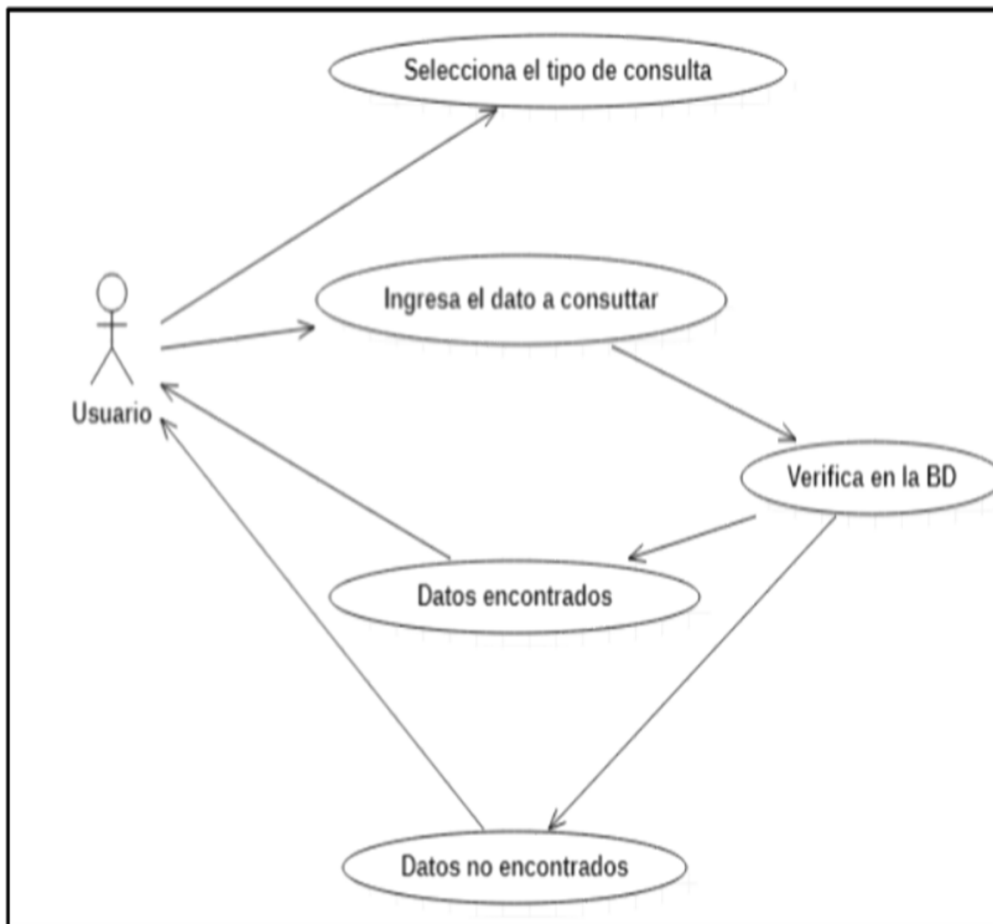
### 3.2.3.6 Consulta de datos

**Objetivo:** presenta la información de acuerdo al requerimiento solicitado por el usuario

**Proceso:**

Selecciona opción

- Ingresa datos a consultar
- Consulta en la base de datos
- Presenta datos encontrados



**Gráfico 9** Consulta de datos

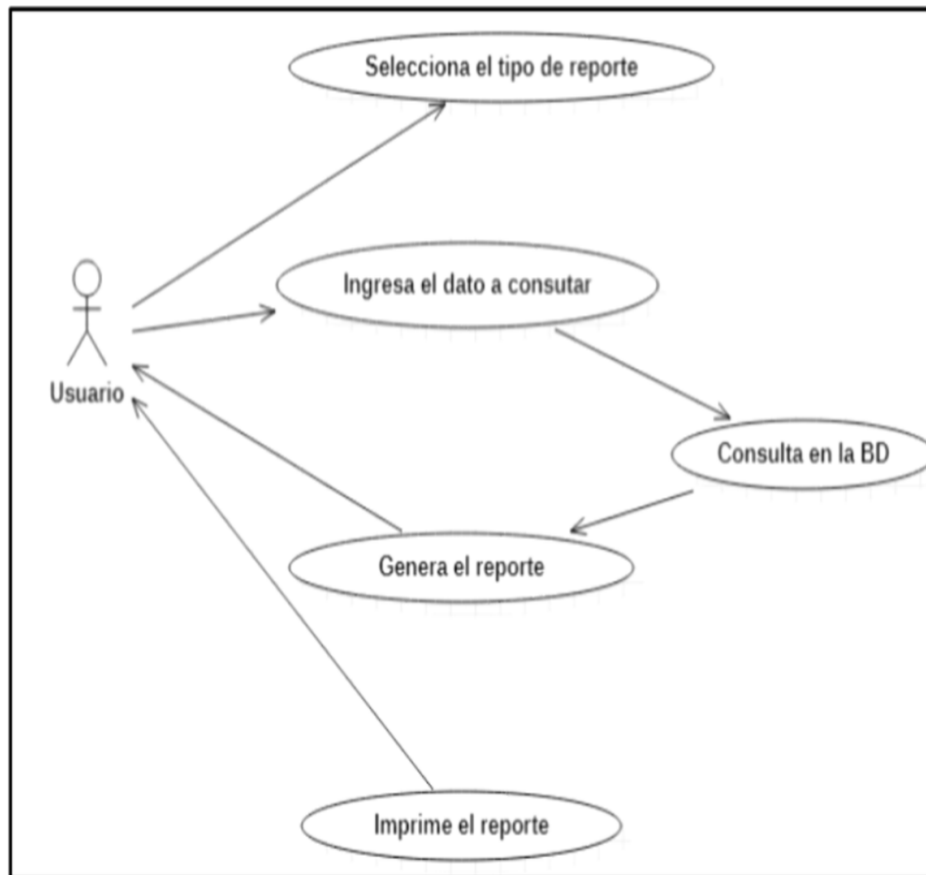
**Fuente:** (Buñay, 2019)

### 3.2.3.7 Reportes de mantenimientos

**Objetivo:** Presenta la información en reportes y la opción de impresión.

**Proceso:**

- Selección de reporte requerido
- Consulta en la Base de Datos
- Presenta Reporte requerido
- Selecciona botón de impresión
- Imprime el reporte



**Gráfico 10** Reportes impresos

**Fuente:** (Buñay, 2019)

### 3.3 DISEÑO

#### 3.3.1 Diseño Conceptual

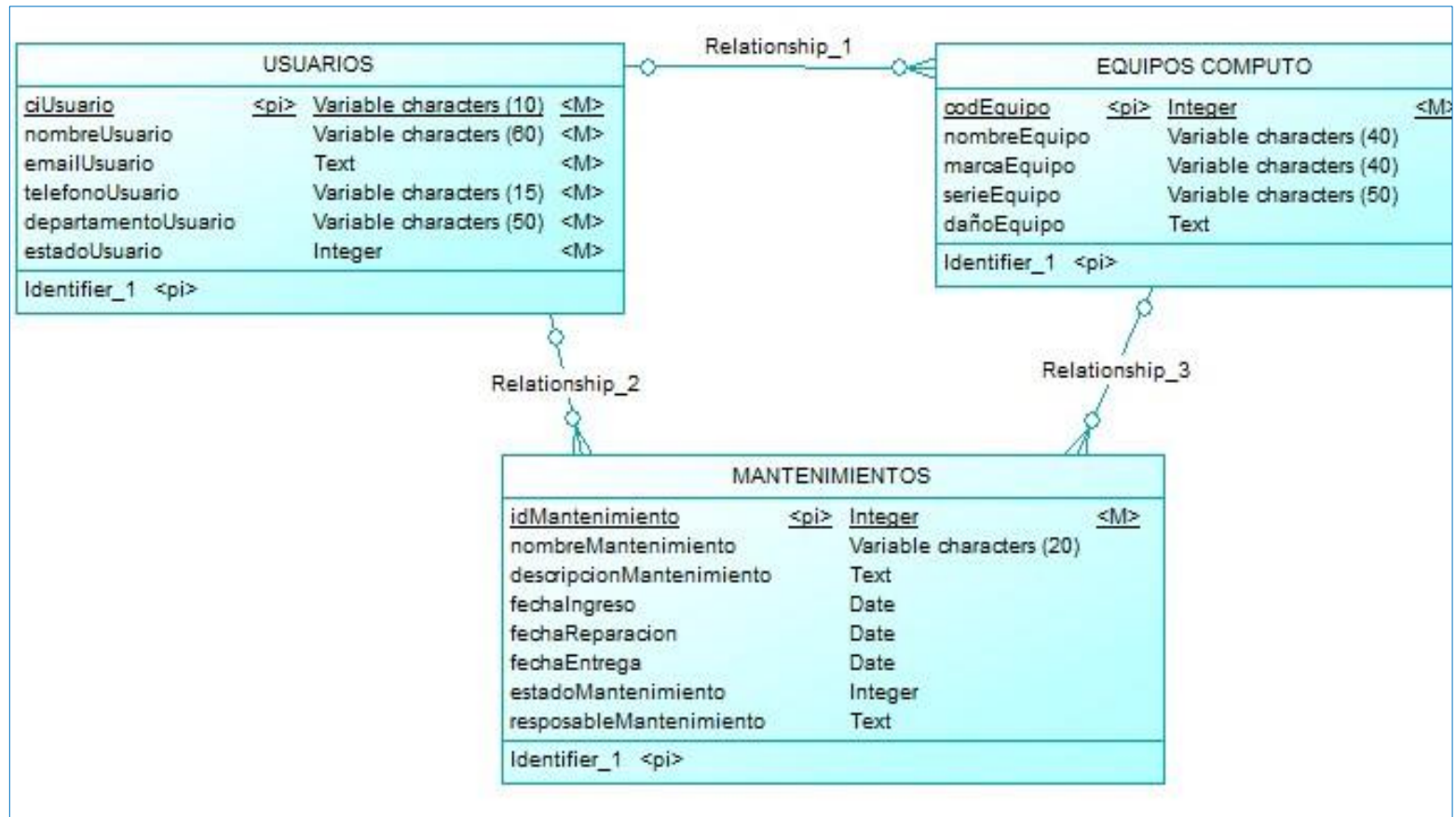


Gráfico 11 Diseño conceptual

Fuente: (Buñay, 2019)

### 3.3.2 Modelo Relacional

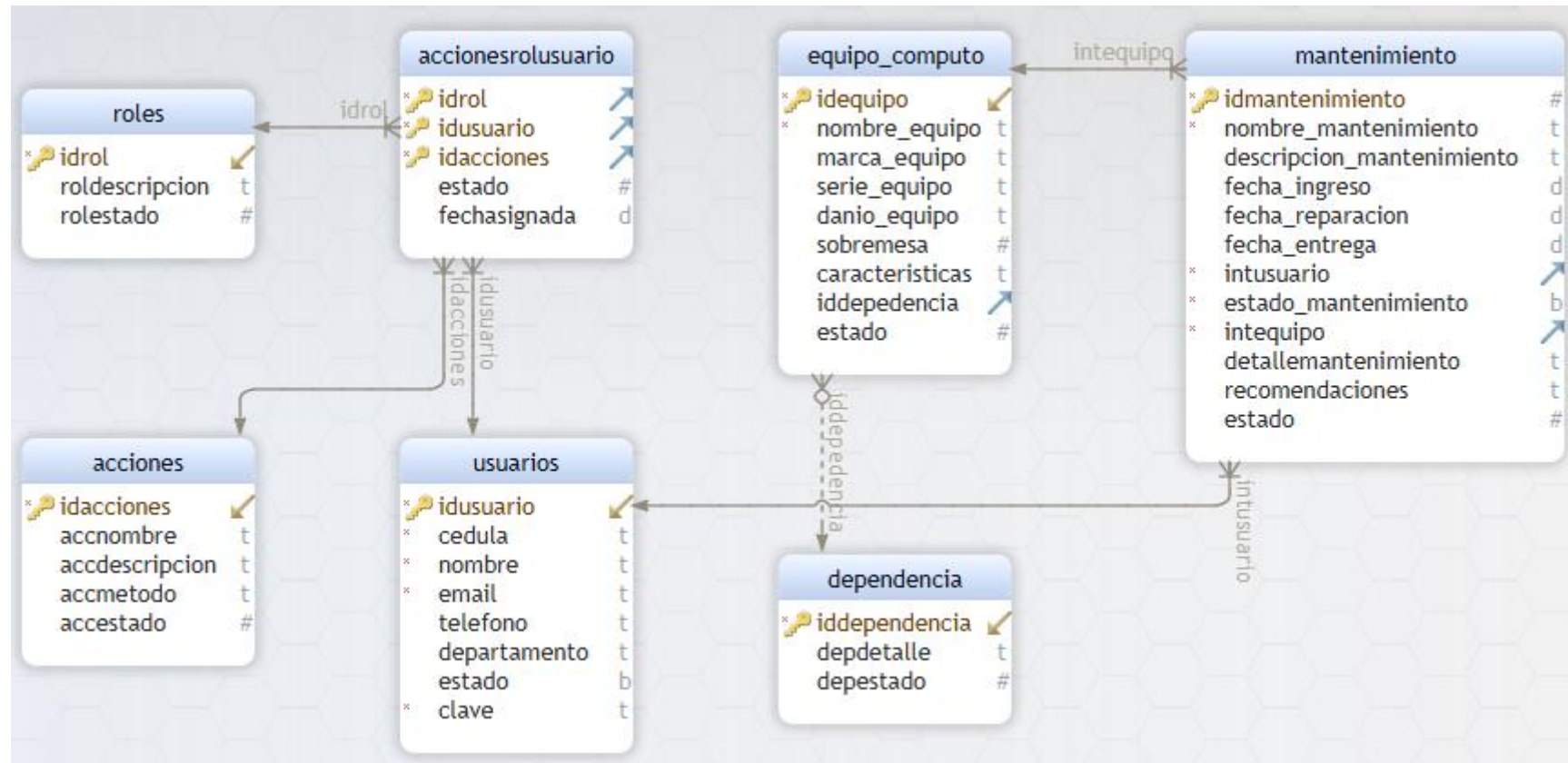


Gráfico 12 Modelo relacional

Fuente: (Buñay, 2019)

### 3.3.3 Diccionario de Datos

Mediante el diccionario de datos se comprenderá de qué tipo son cada campo de la Base de Datos que permitan utilizar en el sistema los datos.

Columna	Tipo	Nulo
PKMANTENIMIENTO	int(11)	No
nombreMantenimiento	varchar(20)	No
descripciónMantenimiento	varchar(200)	No
Fechaingreso	date	No
fecha reparación	date	No
fecha entrega	date	No
estado Mantenimiento	varchar(200)	No
responsable Mantenimiento	varchar(30)	No

**Tabla 8** Mantenimiento

**Fuente:** (Buñay, 2019)

Columna	Tipo	Nulo
PKUSUARIO	int(11)	No
nombreUsuario	varchar(30)	No
emailUsuario	varchar(30)	No
departamentoUsuario	varchar(30)	No
estadousuario	varchar(10)	No
contrasenaUsuario	varchar(10)	No

**Tabla 9** Usuarios

**Fuente:** (Buñay, 2019)



Columna	Tipo	Nulo
PKEQUIPOCOMPUTO	int(11)	No
nombreEquipo	varchar(30)	No
marcaEquipo	varchar(30)	No
serieEquipo	varchar(10)	No
dañoEquipo	varchar(30)	No

**Tabla 10** Registro de Equipo

**Fuente:** (Buñay, 2019)

## Diseño de Interfaces

**Página Principal:** Ingreso de usuario y clave

**Gráfico 13** Pantalla de Login

**Fuente:** (Buñay, 2019)

**Acceso al sistema:** Pantalla principal

Ingreso de datos para un mantenimiento: Ingresar todos los datos requeridos para un mantenimiento

MENU 1	MENU 2	MENU 3	MENU 4	
OPCION 1	OPCION 1	OPCION 1	OPCION 1	
OPCION 2	OPCION 2	OPCION 2	OPCION 2	
OPCION N	OPCION N	OPCION N	OPCION N	
LOGO EMPRESA				

**Gráfico 14** Pantalla Principal

**Fuente:** (Buñay, 2019)

**Ingreso de Datos** Muestra el formulario para el ingreso de datos

INGRESO DE DATOS	
DATO 1	<input type="text"/>
DATO 2	<input type="text"/>
.	
DATO N	<input type="text"/>
<input type="button" value="OPCION 1"/> <input type="button" value="OPCION 2"/>	

**Gráfico 15** Asignación de mantenimiento

**Fuente:** (Buñay, 2019)

**Reporte de Mantenimiento:** Muestra el reporte del mantenimiento

REPORTE DE DATOS				
DATO 1	DATO 2	DATO 3 .....	DATO N	
	OPCION 1		OPCION 2	

**Gráfico 16** Reportes de mantenimiento

**Fuente:** (Buñay, 2019)

**CAPÍTULO IV**  
**IMPLEMENTACIÓN DEL SISTEMA**

## 4.1 Configuración de las herramientas de desarrollo

Para la creación de la aplicación utilizaron múltiples herramientas que se detallarán a continuación:

### 4.1.1 PostgreSQL

Para usar PostgreSQL se necesita instalar:

- Sistema gestor de base de datos
- Una herramienta gráfica para administrar la base de datos. pgAdmin es la herramienta de interfaz gráfica de usuario herramienta más popular para PostgreSQL

Se puede descargar e instalar estos componentes de forma individual, pero combinar las configuraciones entre el servidor de un conjunto de datos pertenecientes a un mismo contexto (bases de datos) y una herramienta de GUI puede ser un desafío. Lo mejor es usar un instalador incluido que se encargue de las complejidades de la configuración.

Se presenta a continuación el proceso que se utiliza para instalar PostgreSQL en una máquina con Windows:

**Paso 1)** direccionar a la página; **Error! Referencia de hipervínculo no válida.** y seleccione Windows

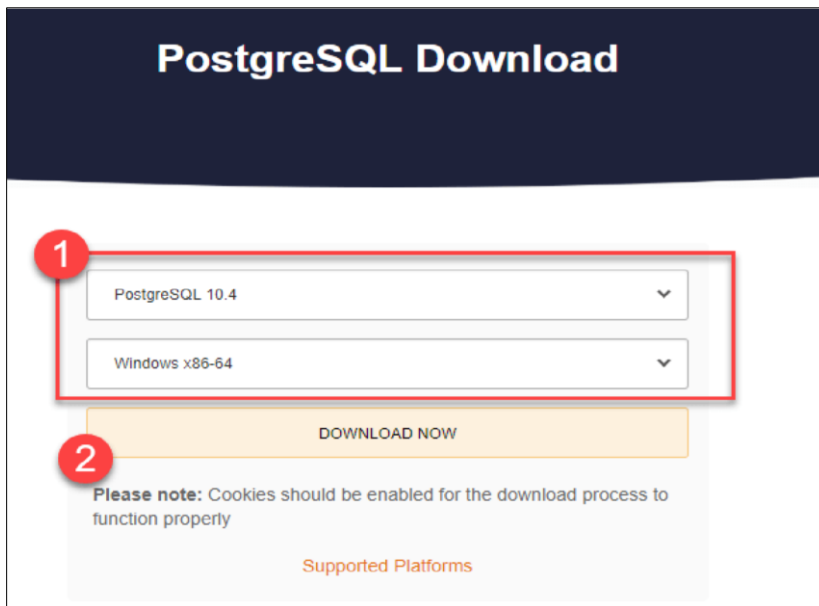
**Paso 2)** existen dos opciones

1. Instalador gráfico por BigSQL
2. Instalador interactivo por EnterpriseDB

**Paso 3)** se pide seleccionar la versión deseada de PostgreSQL y el sistema operativo.

Se debe seleccionar la última versión de PostgreSQL y el sistema operativo según el entorno.

Se hace clic en el botón descargar

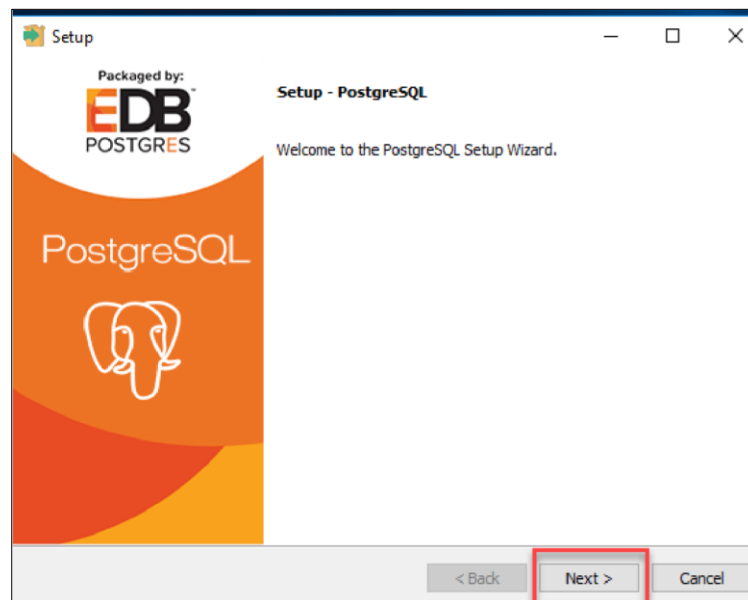


**Gráfico 17** Descarga de PostgreSQL

**Fuente:** (Buñay, 2019)

La descarga comienza.

**Paso 4)** hay que abrir el .exe descargado y hacer clic en siguiente en la pantalla de bienvenida de la instalación.

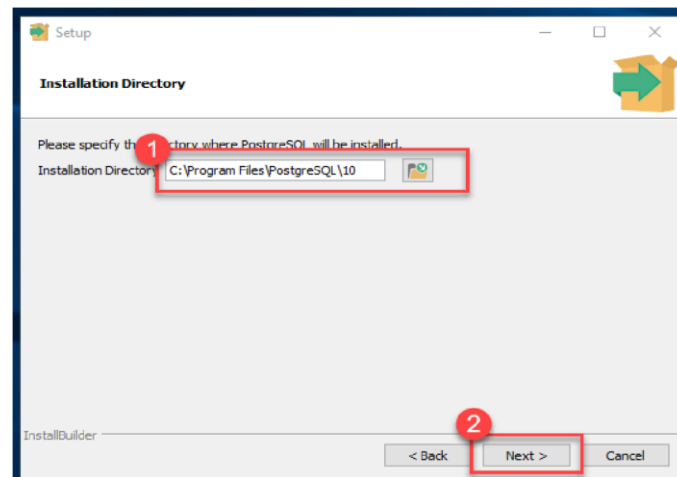


**Gráfico 18** Instalación de PostgreSQL

**Fuente:** (Buñay, 2019)

### Paso 5)

Se puede realizar el cambio de directorio al instalar si fuese necesario, caso contrario existe la posibilidad de dejarlo en predeterminado. Posteriormente, se debe hacer clic en siguiente.

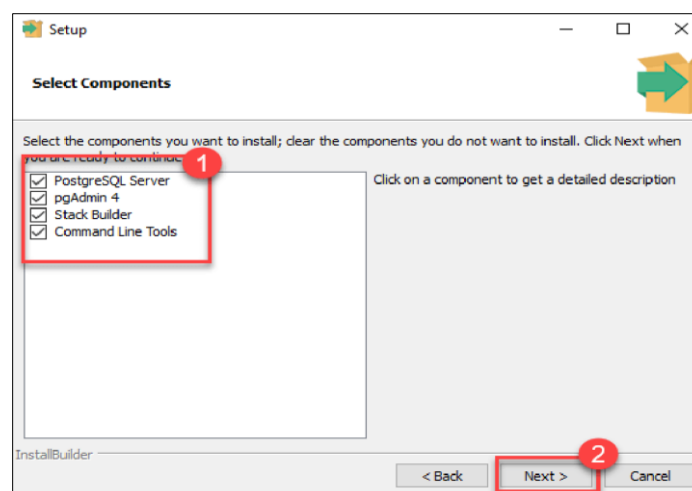


**Gráfico 19** Instalación Directory de PostgreSQL

**Fuente:** (Buñay, 2019)

### Paso 6)

Se puede elegir los componentes que se desea instalar en el sistema. Existe la posibilidad de desmarcar Stack Builder en caso de no desear agregar algunas características. Luego se hace clic en siguiente.



**Gráfico 20** Selección de componentes de PostgreSQL

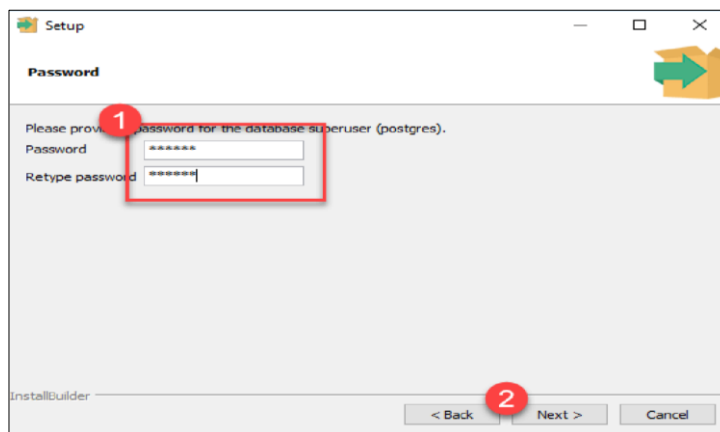
**Fuente:** (Buñay, 2019)

### Paso 7)

Se pueden cambiar la ubicación de los datos y posteriormente se debe hacer clic en siguiente.

### Paso 8)

Hay que ingresar la contraseña de super usuario y posteriormente se debe hacer clic en siguiente.



**Gráfico 21** Password de PostgreSQL

**Fuente:** (Buñay, 2019)

### Paso 9)

Hay que dejar el número de puerto predeterminado y posteriormente se debe hacer clic en siguiente.



**Gráfico 22** Puerto de PostgreSQL

**Fuente:** (Buñay, 2019)



### Paso 10)

Se debe verificar el resumen previo a la instalación y posteriormente se debe hacer clic en siguiente.

### Paso 11)

Se debe hacer clic en siguiente.

### Paso 12)

Finalizada la instalación, se muestra el indicador de Stack Builder, el cual se debe desmarcar. Posteriormente se debe hacer clic en finalizar.

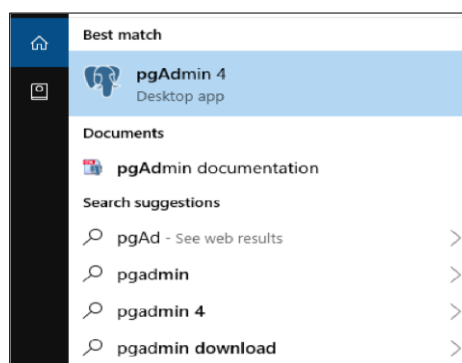


**Gráfico 23** Stack Builder de PostgreSQL

**Fuente:** (Buñay, 2019)

### Paso 13)

Para iniciar PostgreSQL hay que ir al menú de inicio y buscar pgAdmin 4.

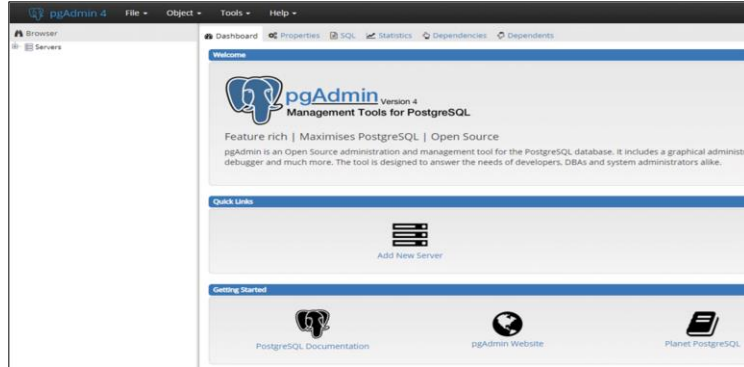


**Gráfico 24** Iniciar PostgreSQL

**Fuente:** (Buñay, 2019)

### Paso 14)

Se observa el inicio de pgAdmin.

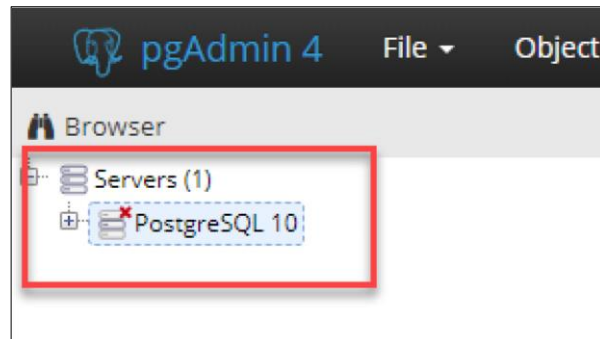


**Gráfico 25** pgAdmin de PostgreSQL

**Fuente:** (Buñay, 2019)

### Paso 15)

Se debe hacer clic en “PostgreSQL 10”.

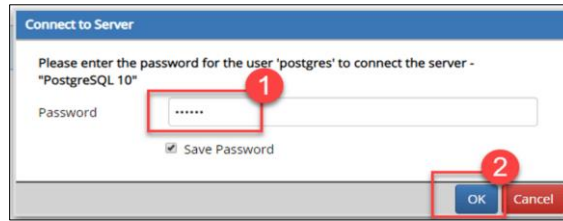


**Gráfico 26** Servidores de PostgreSQL

**Fuente:** (Buñay, 2019)

### Paso 16)

Hay que ingresar la contraseña del SUPERUSUARIO configurada durante la instalación y posteriormente se debe hacer clic en aceptar.

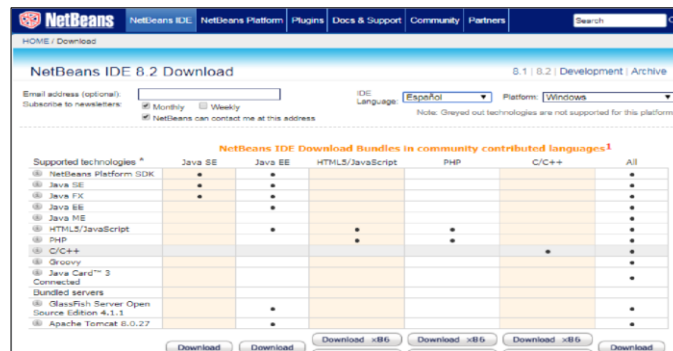


**Gráfico 27** Contraseña de ingreso de PostgreSQL

**Fuente:** (Buñay, 2019)

#### 4.1.2 NetBeans

La aplicación SW free en el internet en la dirección <https://netbeans.org/downloads> fue creada a través de NetBeans.



**Gráfico 28** Página de descarga de NetBeans

**Fuente:** (Buñay, 2019)

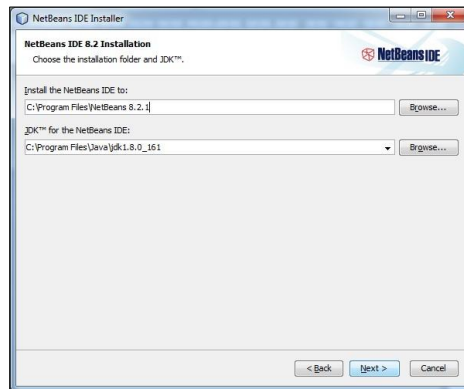
Una vez descargado el programa se ejecutó e inició la instalación.



**Gráfico 29** Ejecución de NetBeans

**Fuente:** (Buñay, 2019)

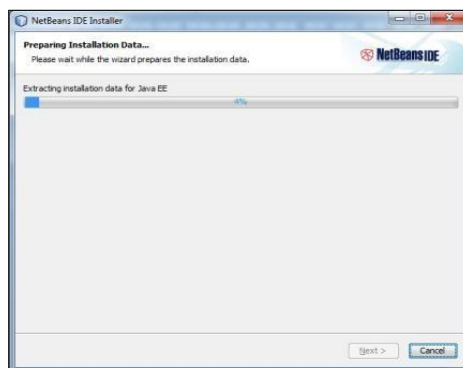
Es importante mencionar que el sistema siempre se va a instalar en el C.



**Gráfico 30** Ubicación de NetBeans

**Fuente:** (Buñay, 2019)

Posteriormente, se continua con la instalación



**Gráfico 31** Instalación de NetBeans

**Fuente:** (Buñay, 2019)

La instalación fue exitosa cuando al finalizar se ingresó al IDE y se empezó con el proyecto.



**Gráfico 32** NetBeans 8.2

**Fuente:** (Buñay, 2019)

## 4.2 Arquitectura del Sistema

La arquitectura del sistema diseñado presenta los siguientes componentes principales:

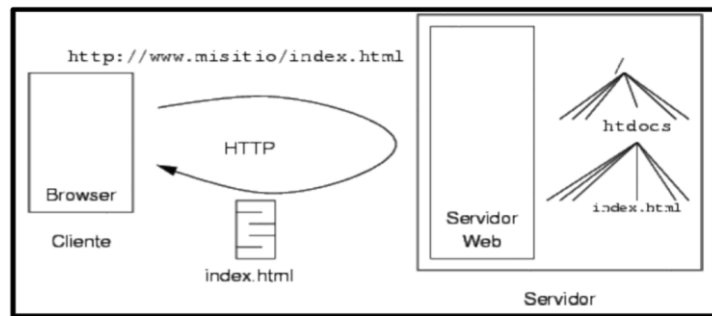
- Un servidor Web
- Una conexión de red
- Uno o más clientes

Las páginas de información formateada a los clientes que las necesitan son distribuidas por el servidor Web.

Los requerimientos son realizados con la utilización de la conexión a la red, utilizando el protocolo HTTP, una vez recibida por el servidor Web, el mismo encuentra la página Web dentro del sistema de archivos y emite de vuelta al navegador que lo requiere.

Hay que manifestar que las aplicaciones Web se enmarcan dentro del modelo Cliente/Servidor mismos que se encargan de gestionar servidores web, y que a su vez emplean como interfaz páginas web. Asimismo, las páginas Web constituyen un componente fundamental de un sitio Web o aplicación. Por ejemplo, los browsers demandan páginas (creadas dinámicamente o almacenadas) con datos importantes a los servidores Web. Con ello, varios de los ambientes de desarrollo, presentan páginas que contienen código scripts dinámicos y HTML, mismos que son ejecutados por medio de los servidores con anterioridad para entregar la página solicitada.

Ante lo expuesto, en la creación del sistema de informático para el registro de mantenimiento de los equipos de cómputo de GADM de Alausí se empleó una arquitectura cliente/servidor. Esta arquitectura permite que el cliente realice las peticiones al sistema que se localiza dentro del servidor el mismo que puede estar en un mismo computador.



**Gráfico 33** Arquitectura Cliente – Servidor

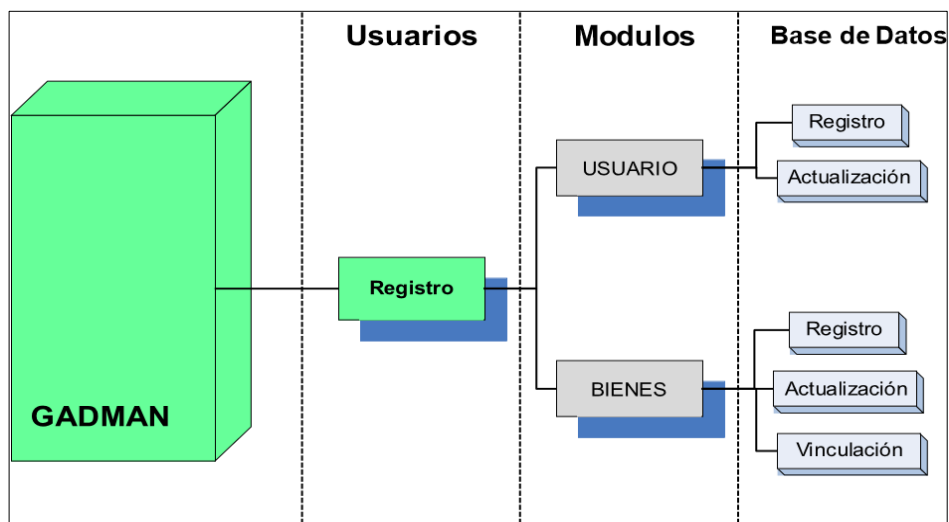
**Fuente:** (Buñay, 2019)

### 4.3 Implementación del Sistema

Se efectuó la implementación del sistema dentro de la institución que lo necesita. Este sistema permite que el usuario y desarrollador tengan una visión completa del mismo. De esta manera, con la implementación, los dos actores mencionados pueden reestructurar los requerimientos en caso de ser necesario.

#### 4.3.1 Definición de Módulos

El sistema consta de dos módulos, los cuales están relacionados en función a los requerimientos del usuario. Se presenta un diagrama de módulos el mismo que permitirá tener un concepto ampliado del sistema:



**Gráfico 34** Definición de módulos del sistema

**Fuente:** (Buñay, 2019)

### 4.3.2 Desarrollo de la aplicación

En la etapa de desarrollo e implementación del sistema se realizó y se pone en práctica, lo expresado con anterioridad en las fases de análisis y diseño, mismo que se muestra dentro del Anexo 1 (Manual de Usuario).

### 4.3.3 Implementación de la aplicación

Para conseguir un mayor beneficio del sistema se requiere de una configuración en los diversos programas que se emplearán en el desarrollo e implementación del sistema especificado anteriormente.

### 4.4 Pruebas de la aplicación

En relación a los elementos funcionales y no funcionales que el usuario entregó para la creación del sistema se realizaron las pruebas, siendo el resultado el siguiente:

Pruebas	Satisfactorio	
	Si	No
Ingreso al sistema	SI	
Validación de contraseña	SI	
Rapidez de consulta	SI	
Disponibilidad de Datos	SI	
Escalabilidad en Sistema	SI	
Seguridad de la información	SI	
Eficiencia en Reportes	SI	
Validación de campos	SI	
Facilidad de uso	SI	
Interfaz intuitiva	SI	

Pruebas	Satisfactorio	
	Si	No
Mensajes de error	SI	
Confirmación de datos correctos	SI	
Registro correcto en la base de datos	SI	
Menú de rápido acceso	SI	

**Tabla 11** Pruebas en el sistema

**Fuente:** (Buñay, 2019)

#### 4.5 Capacitación al Personal

**Facilitador:** Rene Buñay desarrollador del sistema

**Destinatarios:** personal encargado del área de mantenimiento del GAD Alausí.

**Herramientas:** para la capacitación se emplearán varias herramientas tecnológicas como las descritas a continuación.

- Dispositivos electrónicos: computador, proyector de pantalla
- Herramientas tecnológicas: Power Point (diapositivas)

**Tiempo estimado:** 1 hora

#### 4.6 MANTENIMIENTO

Dentro de la implementación del sistema se observan dos casos de mantenimiento:

- **Mantenimiento correctivo:** el cual ayudará al usuario a utilizar el sistema, con lo que el desarrollador podrá corregir errores pasados por alto en el desarrollo.
- **Mantenimiento preventivo:** con el mismo, el desarrollador y el usuario del sistema les permitirá garantizar y prevenir la fiabilidad del sistema.



## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

## 5.1 Conclusiones

- El IDE - NetBeans permite la ejecución en diferentes lenguajes de programación, siendo Java la herramienta más potente para la creación de aplicaciones web plataformas y su compatibilidad con el motor de base de datos PostgreSQL.
- El análisis de los procesos internos y normativa del GADM del cantón Alausí facilitó la elaboración del prototipo del Sistema web para el control y registro de los equipos de cómputo de la Institución.
- El diseño de la base de datos en la herramienta Star - UML facilitó la interpretación de la arquitectura del sistema para el almacenamiento de información de la institución.
- Con la implementación del sistema web GADMANT, se logró tener el control total de los registros de mantenimiento de los equipos de cómputo que existen en el Municipio de Alausí.
- Las pruebas de funcionamiento y optimización del GADMANT permiten comprobar y almacenar los registros de mantenimiento a los equipos de cómputo del GADM Alausí y no tener pérdidas de información a futuro.

## **5.2. Recomendaciones**

A la unidad de Tecnología y Comunicación del GAD Municipal de Alausí.

- Realizar cambios del Sistema Web a nivel de base de datos y en la interfaz de acuerdo a las nuevas necesidades, este proceso será de fácil desarrollo gracias a la escalabilidad que posee el sistema y a la ayuda que se ofrece en el manual técnico.
- Realizas capacitaciones continuas a los nuevos practicantes en la utilización del sistema informático, con la finalidad de mantener activo el sistema.

## BIBLIOGRAFÍA

- Chacon. (12 de Enero de 2017). Sistemas Informaticos. Obtenido de <https://www.preparadores.eu/temamuestra/PTecnicos/PComerciales.pdf>
- Adriana Gissela, C. L. (2018). Sistema informático de enfermería para vigilancia del esquema de inmunizaciones a menores de dos años, noviembre 2017 – febrero 2018. Universidad Nacional de Chimborazo, 2018. Obtenido de <http://dspace.unach.edu.ec/handle/51000/4790>
- Colcha, V. (6 de 6 de 2015). Análisis Comparativo entre los Motores de Base de Datos Postgresql y Firebird Aplicando al Gobierno Autónomo Descentralizado Municipal de San José de Chimbo. Escuela Superior Politécnica de Chimborazo. Obtenido de <http://dspace.esPOCH.edu.ec/handle/123456789/4385>
- Delgado, J., Balbuca, D., & Ortiz, J. (2017). Análisis comparativos de servidores de aplicaciones open SOURCE para la plataforma JAVA EE. caso práctico: módulo de gestión de juntas administradoras de agua potable y riego para la dirección provincial de Chimborazo de la Senagua. Universidad Nacional de Chimborazo, 2017. Obtenido de <http://dspace.unach.edu.ec/handle/51000/3397>
- Freddy, C. (20 de 2 de 2020). Desarrollo de una aplicación web para el control de permisos de funcionamiento del cuerpo de bomberos del cantón Tisaleo utilizando Java EE Y PostgreSQL. Escuela Superior Politécnica de Chimborazo. Obtenido de <http://dspace.esPOCH.edu.ec/handle/123456789/13939>
- Llanga, A., Haro, S., & Inca, L. (2016). “Diseño e implementación un sistema de telecomunicación para los dispositivos electrónicos de personas con discapacidades de la universidad nacional de Chimborazo, que permita ayudar en la orientación, ubicación e interacción entre los usuarios y la socie. Riobamba: Universidad Nacional de Chimborazo, 2016. Obtenido de <http://dspace.unach.edu.ec/handle/51000/2877>
- Lovato, L. (2018). Desarrollo e implementación de un sistema informático para el control de las comisiones vehiculares en el comando de apoyo logístico n° 11

“Calicuchima” agregado a la brigada de Caballería Blindada n° 11 “Galápagos” realizado en java y gestor de base de d. Instituto Tecnológico Superior “San Gabriel” 2018. Obtenido de <https://sangabrielriobamba.edu.ec/tesis/sistemas/tesis010.pdf>

Núñez Zamora, W. R. (8 de agosto de 2015). Análisis Comparativo de la Productividad de las Herramientas para Realidad Aumentada Vuforia y Layar del Sistema de Información Móvil de Infraestructura Física de la Epoch. Riobamba: Escuela Superior Politécnica de Chimborazo. Obtenido de <http://dspace.esepoch.edu.ec/handle/123456789/4379>

Ortiz Quishpe, L. M. (21 de marzo de 2016). Sistema Informático de Seguimiento de la Información Docente Ocasional y Titular de la Facultad de Informática y Electrónica. (15), 1-30. Escuela Superior Politécnica de Chimborazo. Obtenido de <http://dspace.esepoch.edu.ec/handle/123456789/4745>

Samantha, T. (2018). Diseño e implementación de un sistema para el inventario y registro de mantenimiento de los equipos informáticos del instituto tecnológico superior san gabriel desarrollado en java y gestor de base de datos mysql. Obtenido de <https://sangabrielriobamba.edu.ec/tesis/sistemas/tesis011.pdf>

Abd-Alrazaq, A. A., Alajlani, M., Alhuwail, D., Erbad, A., Giannicchi, A., Shah, Z., ... & Househ, M. (2021). Blockchain technologies to mitigate COVID-19 challenges: A scoping review. *Computer methods and programs in biomedicine update*, 1, 100001.

Ardian, D., Larasati, P. D., & Irawan, A. (2018). Perancangan sistem informasi aplikasi rental mobil menggunakan java netbeans dan MySQL pada perusahaan Dean's Car Rent. *Applied Information System and Management (AISM)*, 1(1), 35-44.

Arzoo, M. K., Prof, A., & Rathod, K. (2017). K-Means algorithm with different distance metrics in spatial data mining with uses of NetBeans IDE 8. 2. *Int. Res. J. Eng. Technol*, 4(4), 2363-2368.

Beyer, D. (2019, April). Automatic verification of C and Java programs: SV-COMP 2019. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (pp. 133-155). Springer, Cham.

- Brown, N. C., & Altadmri, A. (2017). Novice Java programming mistakes: Large-scale data vs. educator beliefs. *ACM Transactions on Computing Education (TOCE)*, 17(2), 1-21.
- Cahan, A., & Cimino, J. J. (2017). A learning health care system using computer-aided diagnosis. *Journal of medical Internet research*, 19(3), e6663.
- Chen, H., Wen, Y., Zhu, M., Huang, Y., Xiao, C., Wei, T., & Hahn, A. (2021). From automation system to autonomous system: An architecture perspective. *Journal of Marine Science and Engineering*, 9(6), 645.
- Cruttwell, G., Gallagher, J., & Pronk, D. (2021). Categorical semantics of a simple differential programming language. *arXiv preprint arXiv:2101.10491*.
- Denby, B., & Lucia, B. (2020, March). Orbital edge computing: Nanosatellite constellations as a new class of computer system. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems* (pp. 939-954).
- Dhika, H., Isnain, N., & Tofan, M. (2019). Manajemen Villa Menggunakan Java Netbeans Dan Mysql. *IKRA-ITH INFORMATIKA: Jurnal Komputer Dan Informatika*, 3(2), 104-110.
- Edwards, S. H., Kandru, N., & Rajagopal, M. B. (2017, August). Investigating static analysis errors in student Java programs. In *Proceedings of the 2017 ACM Conference on International Computing Education Research* (pp. 65-73).
- Gómez, Á. (2022). *Auditoría de seguridad informática*. Ediciones de la U.
- Greenbaum, J., & Kyng, M. (Eds.). (2020). *Design at work: Cooperative design of computer systems*. CRC Press.
- Guo, H. (2018). Application of a Computer-Assisted Instruction System Based on Constructivism. *International Journal of Emerging Technologies in Learning*, 13(4).

- Habrat, M., Krakowska, P., Puskarczyk, E., Jędrychowski, M., & Madejski, P. (2017). The concept of a computer system for interpretation of tight rocks using X-ray computed tomography results. *Studia Geotechnica et Mechanica*, 39(1).
- Eck, D. (2021). *Introduction to programming using Java*. Hobart and William Smith Colleges.
- Katona, J., & Kovari, A. (2018). Examining the learning efficiency by a brain-computer interface system. *Acta Polytechnica Hungarica*, 15(3), 251-280.
- Kontopantelis, E., Stevens, R. J., Helms, P. J., Edwards, D., Doran, T., & Ashcroft, D. M. (2018). Spatial distribution of clinical computer systems in primary care in England in 2016 and implications for primary care electronic medical record databases: a cross-sectional population study. *BMJ open*, 8(2), e020738.
- Lopez, D. A. P. (2021). Requirement Analysis for Your Full-Stack Web Application. In *Full-Stack Web Development with Jakarta EE and Vue.js* (pp. 111-142). Apress, Berkeley, CA.
- Martínez, J. C., Henao, C., Henao, F., & Zapata, E. (2021). Utilización de Arquitecturas Limpias para Trabajo con Buenas Prácticas en la Construcción de Aplicaciones Java. *Revista Innovación Digital y Desarrollo Sostenible-IDS*, 1(2), 133-140.
- Psarommatis, F., & Kiritsis, D. (2022). A hybrid Decision Support System for automating decision making in the event of defects in the era of Zero Defect Manufacturing. *Journal of Industrial Information Integration*, 26, 100263.
- Roffel, B., & Chin, P. (2017). *Computer control in the process industries*. CRC Press.
- Rumyancev, I. A., Ivanov, M. A., & Aivazova, M. O. (2021, January). Multi-user system for manual and automated online measurements. In *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)* (pp. 1051-1054). IEEE.
- Saleh, H., Zykov, S., & Legalov, A. (2021). Eolang: Toward a new java-based object-oriented programming language. In *Intelligent Decision Technologies* (pp. 355-363). Springer, Singapore.

- Santos, M. A., Meerhof, D., Quiroga, V., Vazquez, C., & Viera, O. (2019). TICs aplicada a contextos sociales críticos: proyecto Nuevo Amanecer. In *VI Simposio Argentino sobre Tecnología y Sociedad (STS 2019)-JAIIO 48 (Salta)*.
- Savitch, W. (2019). *Java: An introduction to problem solving & programming*. Pearson Education Limited.
- Shan, G., Zheng, Y., Xing, C., Chen, D., Li, G., & Yang, Y. (2022). Architecture of Computing System based on Chiplet. *Micromachines*, *13*(2), 205.
- Singh, N., Chouhan, S. S., & Verma, K. (2021, October). Object oriented programming: Concepts, limitations and application trends. In *2021 5th International Conference on Information Systems and Computer Networks (ISCON)* (pp. 1-4). IEEE.
- Soares, F. M., & Souza, A. M. (2017). *Neural network programming with Java*. Birmingham, UK: Packt publishing.
- Souza-Pereira, L., Ouhbi, S., & Pombo, N. (2021). Quality-in-use characteristics for clinical decision support system assessment. *Computer Methods and Programs in Biomedicine*, *207*, 106169.
- Tabares Quiroz, J., Correa Vélez, S., & Herrera Caballero, J. M. (2021). Metamorfosis organizacionales del cambio tecnológico: integración de sistemas informáticos en una organización de salud colombiana. *Innovar*, *31*(79), 103-116.
- Viloria, A., Acuña, G. C., Franco, D. J. A., Hernández-Palma, H., Fuentes, J. P., & Rambal, E. P. (2019). Integration of data mining techniques to PostgreSQL database manager system. *Procedia Computer Science*, *155*, 575-580.
- White, G. B., Fisch, E. A., & Pooch, U. W. (2017). *Computer system and network security*. CRC press.
- Wijaya, K. (2019). Implementasi Metode Prototype Dalam Rancang Bangun Sistem Informasi Perpustakaan Menggunakan Java (Netbeans 7.3)(Studi Kasus SMK N 01 Prabumulih). *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, *8*(1), 53-60.



Woltmann, L., Olwig, D., Hartmann, C., Habich, D., & Lehner, W. (2021). PostCENN: postgresql with machine learning models for cardinality estimation. *Proceedings of the VLDB Endowment*, 14(12), 2715-2718.

Yuan, Y., & Banzhaf, W. (2018). Arja: Automated repair of java programs via multi-objective genetic programming. *IEEE Transactions on software engineering*, 46(10), 1040-1067.

## **ANEXOS**


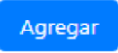


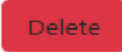
**ANEXO 1:**  
**MANUAL DEL USUARIO**

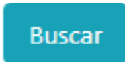
## 1. OBJETO DEL DOCUMENTO

El presente documento pretende mostrar al usuario el funcionamiento del sistema diseñado para el GADM de Alausí, con lo cual se pretende mostrar el funcionamiento de la aplicación de una manera clara y concisa.

## 2. MANUAL DE USUARIO

### 2.1. BOTONES

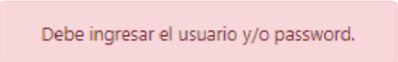

Botón	Descripción
	Validar el tipo de usuario y contraseña e ingresar al sistema
	Agregar nuevos datos a un registro.
	Actualizar los datos del registro seleccionado
	Editar los datos del registro seleccionado
	Eliminar los datos del registro seleccionado

Botón	Descripción
	Realiza una búsqueda de información según el criterio especificado para la búsqueda

**Tabla N° 1.** Botones del sistema

Elaborado por: Rene Buñay

## MENSAJES

Pantallas	Descripción
	Cuando no se han ingresado el nombre de usuario y contraseña
	Cuando se ha ingresado mal el nombre de usuario y contraseña.

**Tabla N° 2.** Mensajes del sistema

Elaborado por: Rene Buñay

## 2.2. INGRESO AL SISTEMA

El usuario debe estar registrado en el sistema con anterioridad. Porque, si no es el caso de que este registrado no le permitirá el ingreso, esto garantiza la confidencialidad de los datos.

Bienvenidos al Sistema

Iniciar Sesión



Usuario:

0603519372

Password:

\*\*\*\*\*

Ingresar

Copyright © 2020 GADM ALAUSÍ. Todos los derechos reservados.

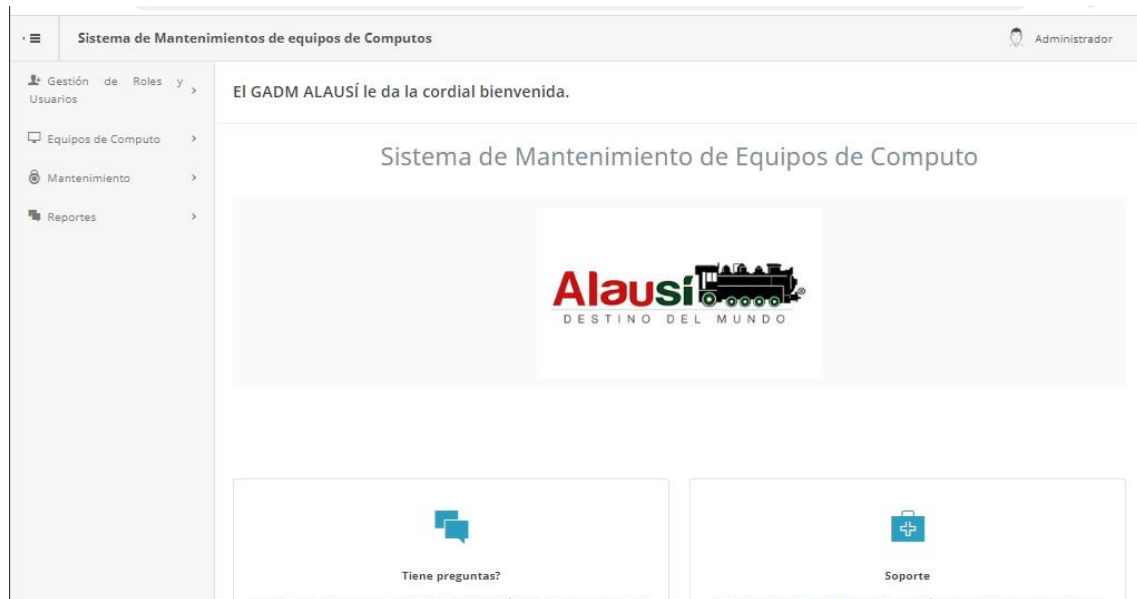
## Gráfico N° 1. Login

Elaborado por: Rene Buñay

### 2.3. PANTALLA PRINCIPAL

Se muestra todas las opciones que permitirán realizar tareas más rápidas y precisas dentro del sistema:

- Inicio
- Equipo de cómputo
- Usuario
- Mantenimiento
- Reportes



**Gráfico N° 2. Pantalla principal**

**Elaborado por: Rene Buñay**

## 2.4. EQUIPOS DE CÓMPUTO

N°	Nombre	Marca	Serie	Año	Dependencia	Estado	Acciones
1	HP probobook 4520s	HP	5CG648253T	2020	TALENTO HUMANO	Habilitado	
2	Cpu Super Power	HP	5CG689253T	2015	UNIDAD DE AMBIENTE	Habilitado	
3	Cpu Super Power	CPU	9CG648258Y	2017	FINANCIERO	Habilitado	
4	Lenovo G4	Lenovo	5CG648258Y	2015	OBRAS PÚBLICAS	Habilitado	
6	HP Proliant ML350G6	HP	6CG648253D	2018	UNIDAD DE DESARROLLO SOCIA	Habilitado	
7	ACER	ACER	1410107A0460	2017	UNIDAD DE AMBIENTE	Habilitado	
8	HP Probook 640	HP	1410107A0031	2018	COMISARIO MUNICIPAL Y DE CONSTRUCCIONES	Habilitado	
9	Computador QBEX	Lenovo	1410107A0345	2019	ADMINISTRACIÓN DE BIENES	Habilitado	
10	HP ProBook 6470b	HP	1410R07033T	2018	CATASTRO	Habilitado	
11	Portátil DELL	DELL	1410107A0286	2019	UNIDAD DE AMBIENTE	Habilitado	

**Gráfico N° 3. Registro de Equipos de Cómputo**

**Elaborado por: Rene Buñay**

## 2.5. USUARIOS

Cuando se requiere de un nuevo accesorio, se ingresa el nombre con su respectiva descripción.

N°	Nombre	Cedula	Departamento	Telefono	Email	Estado	Acciones
1	RENE BUÑAY	0605163591	TECNICO	0968040841	rene.b@hotmail.com	Habilitado	[Iconos]
2	Administrador	admin	SISTEMAS	0984956372	administrador@gadmlausi.com	Desabilitado	[Iconos]
3	SISLEMA MENDOZA JOSE LUIS	0603361866	JEFE DE UTICS	0995684281	sislema-luis98@gmail.com	Habilitado	[Iconos]

Gráfico N° 4. Registro de Usuarios

Elaborado por: Rene Buñay

## 2.6. MANTENIMIENTO

Permite ingresar los bienes en el registro para tener un mejor control.

N°	Nombre Mantenimiento	Fecha	Estado Mantenimiento	Equipo	Responsable	Estado	Acciones
1	Mantenimiento de hardware	Tue Jun 07 00:00:00 COT 2022	Entregado	HP probook 4520s	SISLEMA MENDOZA JOSE LUIS	Habilitado	[Iconos]
2	Comprobación de la memoria RAM y disco duro	Wed Jun 08 00:00:00 COT 2022	Entregado	Lenovo G40	SISLEMA MENDOZA JOSE LUIS	Habilitado	[Iconos]
3	Mantenimiento de software	Tue Jun 07 00:00:00 COT 2022	Entregado	Cpu Super Power	SISLEMA MENDOZA JOSE LUIS	Habilitado	[Iconos]
4	Computador con contraseña de Microsoft	Tue Jun 07 00:00:00 COT 2022	Entregado	Lenovo G4	SISLEMA MENDOZA JOSE LUIS	Habilitado	[Iconos]
5	Computador con contraseña de usuario	Tue Jun 07 00:00:00 COT 2022	Entregado	Cpu Super Power	SISLEMA MENDOZA JOSE LUIS	Habilitado	[Iconos]

Gráfico N° 5. Registro de Mantenimientos

Elaborado por: Rene Buñay



## 2.7. REPORTES

Permite generar tres tipos de reportes:

- Mantenimiento por equipo
- Mantenimiento por usuario
- Número de mantenimientos

N°	Nombre Mantenimiento
1	Mantenimiento hardware
2	Comprobación de memoria RAM y disco duro

**Gráfico N° 6.** Reportes

**Elaborado por:** Rene Buñay

### 2.7.1. REPORTES - Mantenimiento por equipo

Permite generar reportes seleccionando el nombre del equipo de cómputo, de igual manera admite imprimir el reporte utilizando (CTL+P).



**Gráfico N° 7. Reportes**

**Elaborado por: Rene Buñay**

### 2.7.2. REPORTES - Mantenimiento por usuario

Permite generar reportes seleccionando el nombre del usuario, de igual manera admite imprimir el reporte.



**Gráfico N° 8. Reportes**

**Elaborado por: Rene Buñay**

### 2.7.3. REPORTES – Número de Mantenimientos

Permite crear un reporte general de todos los mantenimientos ejecutados, de igual forma admite imprimir el reporte.

The screenshot displays a web application interface for a computer equipment maintenance system. The main content area shows a report titled "REPORTE TOTAL DE MANTENIMIENTOS" with a table of maintenance records. The table has the following data:

Nombre	Estado	Fecha Ingreso	Fecha Entrega	Equipo	Dependencia	Responsable
Mantenimiento de hardware	Entregado	7 Jun 2022	19 Jan 2019	HP probook 4520s	TALENTO HUMANO	SISLEMA MENDOZA JOSE LUIS
Comprobación de la memoria RAM y disco duro	Entregado	8 Jun 2022	1 Feb 2022	Lenovo G40	TALENTO HUMANO	SISLEMA MENDOZA JOSE LUIS
Mantenimiento de software	Entregado	7 Jun 2022	12 Jan 2021	Cpu Super Power	FINANCIERO	SISLEMA MENDOZA JOSE LUIS
Computador con contraseña de Microsoft	Entregado	7 Jun 2022	4 Jan 2021	Lenovo G4	OBRAS PUBLICAS	SISLEMA MENDOZA JOSE LUIS
Computador con contraseña de usuario	Entregado	7 Jun 2022	6 Jan 2021	Cpu Super Power	UNIDAD DE AMBIENTE	SISLEMA MENDOZA JOSE LUIS

**Gráfico N° 9. Reportes**  
**Elaborado por: Rene Buñay**

**ANEXO 2:**  
**MANUAL DEL TÉCNICO**

## 1. Requerimiento del sistema

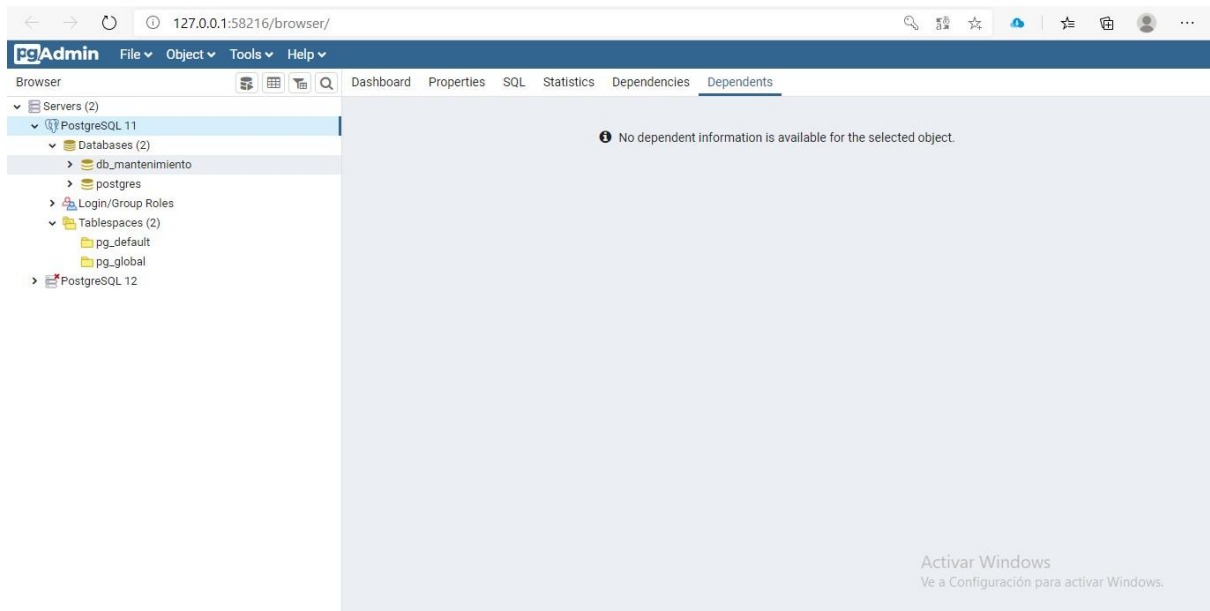
Software utilizado para la creación de la aplicación:

- WINDOWS 10
- NETBEANS IDE 8.2
- POSTGRESQL
- JAVA SPLASK SCREEN

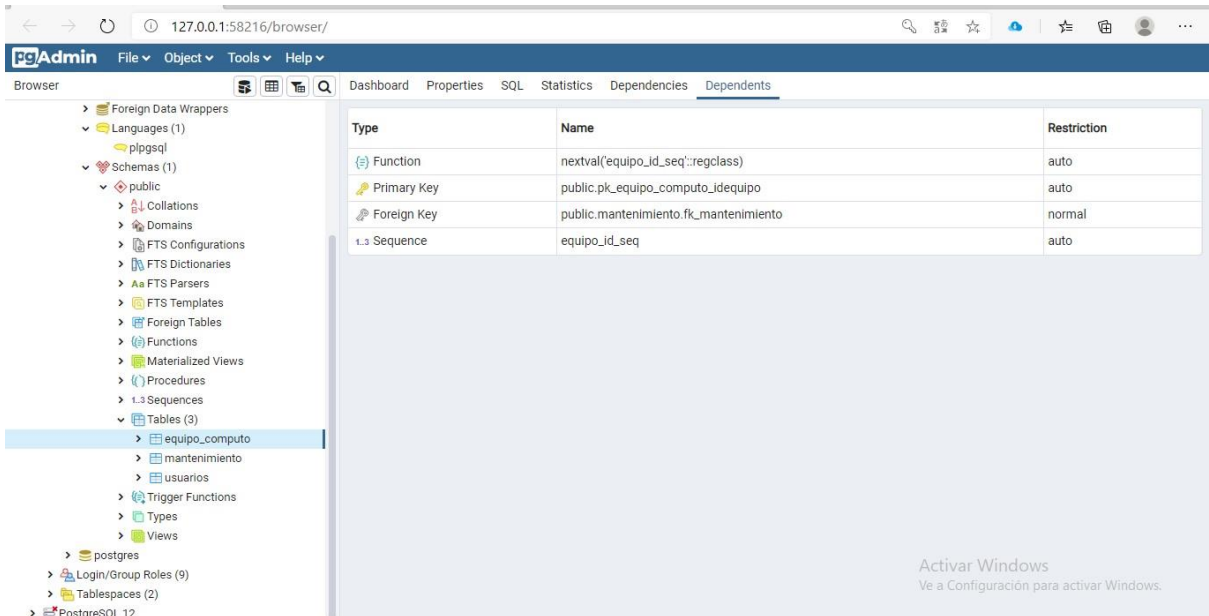
## 2. Crear conexión con la base de datos PostgreSQL



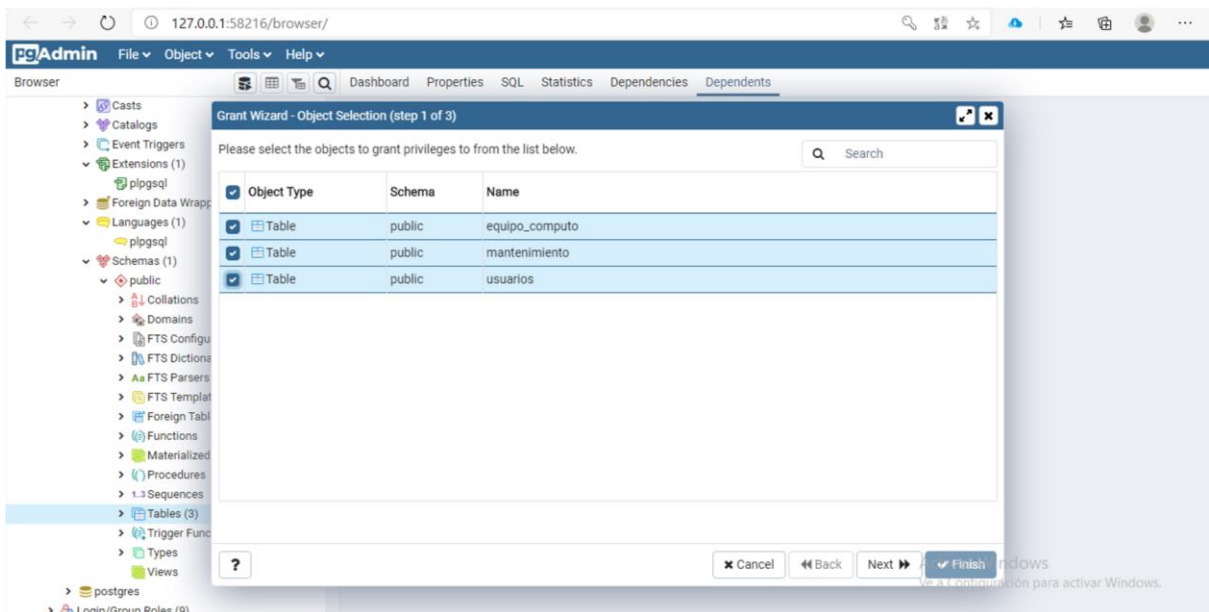
## 3. Creación de la base de datos mantenimiento



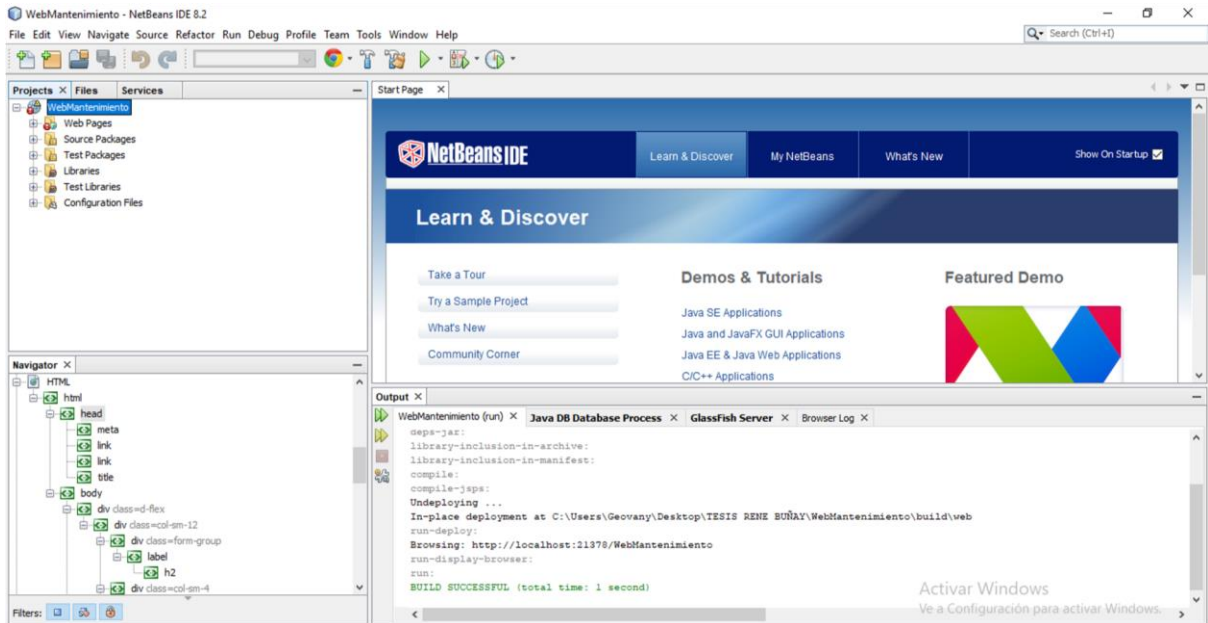
## 4. Creación de las tablas



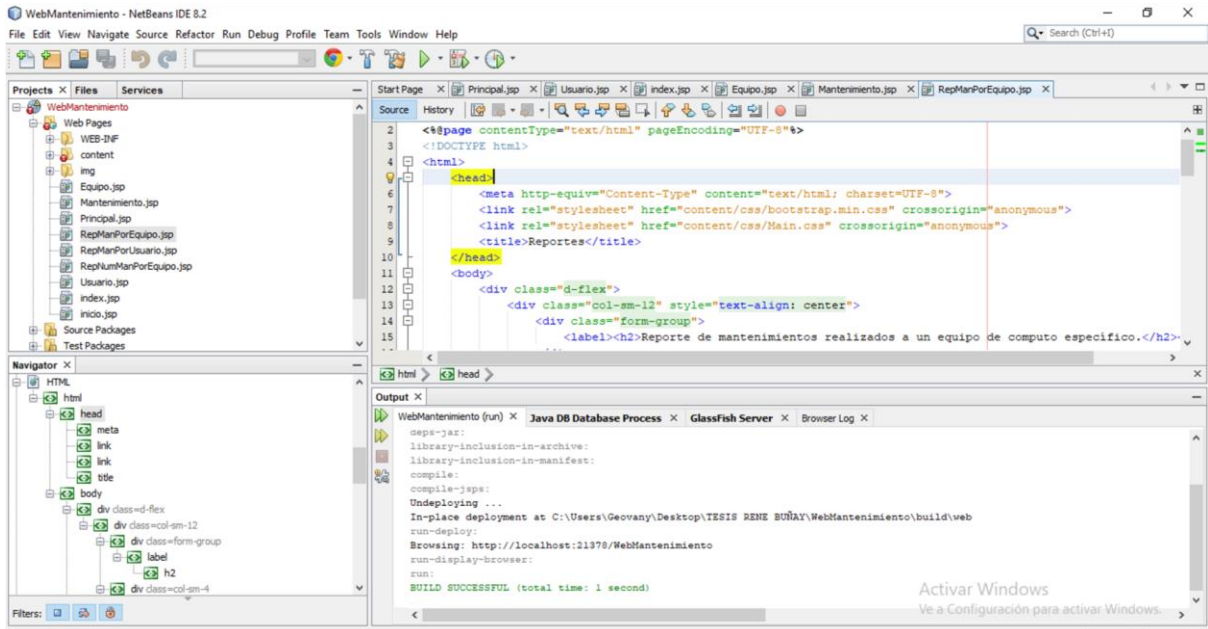
## 5. Creación de campos y manipulación de registros



## 6. Creación de la aplicación con NetBeans 8.2



## 7. Codificación de interfaces



## 8. Index

Este código que se muestra a continuación es en HTML, el lenguaje que interpreta el computador para mostrar una página web.

**Nombre archivo:** Index.jsp

El archivo que contine este código fuente se encuentra en el servidor en la siguiente ruta:

**D:\payara41\domains1\sistemas\matenimientoIU\src\main\webapp\Index.jsp**

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" href="content/css/bootstrap.min.css"
crossorigin="anonymous">
    <link rel="stylesheet" href="content/css/Main.css" crossorigin="anonymous">
    <title>Mantenimiento</title>
  </head>
  <body>
    <div class="container mt-4 col-lg-4">
      <div class="card col-sm-10">
        <div class="card-body">
          <form class="form-sign" action="Validar" method="POST">
            <div class="form-group text-center">
              <h3>Iniciar Sesión</h3>
              
              <label>Bienvenidos al Sistema</label>
              <c:choose>
                <c:when test="${(cond == 1)}">
                  <div class="alert alert-danger alert-dismissible">Debe ingresar el
usuario y/o password.</div>
                </c:when>
              </c:choose>
            </div>
          </form>
        </div>
      </div>
    </div>
  </body>
</html>
```



```

        <c:when test="\${(cond == 2)}">
            <div class="alert alert-danger alert-dismissible">Credenciales
incorrectas.</div>
        </c:when>
        <c:otherwise>
            </c:otherwise>
        </c:choose>
    </div>
    <div class="form-group">
        <label>Usuario:</label>
        <input type="text" name="txtuser" value="" class="form-control"
placeholder="Cédula">
    </div>
    <div class="form-group">
        <label>Password:</label>
        <input type="password" name="txtpass" value="" class="form-
control">
    </div>
    <input type="submit" name="accion" value="Ingresar" class="btn btn-
primary btnblock">
    </form>
</div>
</div>
<script src="content/js/jquery-3.3.1.slim.min.js"
crossorigin="anonymous"></script>
<script src="content/js/popper.min.js" crossorigin="anonymous"></script>
<script src="content/js/bootstrap.min.js" crossorigin="anonymous"></script>
<script src="content/js/funciones.js" crossorigin="anonymous"></script>
</body>
</html>

```

## 9. Menú Principal

Aquí se listan las opciones que tiene el menú del sistema, cabe mencionar que cada una de las opciones están de acuerdo al rol y permisos que han configurado anteriormente.

**Nombre archivo:** MenuFuncionario.jsp

El archivo que contine este código fuente se encuentra en el servidor en la siguiente ruta:

**D:\payara41\domains1\sistemas\matenimientoIU\src\main\webapp\MenuFuncionario.jsp**

```
<% @page import="entidadesMa.Roles"%>
<% @page import="javax.persistence.Persistence"%>
<% @page import="entidadesMa.Accionesrolusuario"%>
<% @page import="javax.naming.InitialContext"%>
<% @page import="controllers.AccionesrolusuarioJpaController"%>
<% @page import="javax.transaction.UserTransaction"%>
<% @page import="javax.persistence.EntityManagerFactory"%>
<% @page import="java.util.List"%>
<% @page import="java.text.SimpleDateFormat"%>
<% @page import="java.util.ArrayList"%>
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
    <!--<meta http-equiv="Content-Security-Policy" content="upgrade-insecure-requests">-->

    <link rel="shortcut icon" href="images/favicon.ico">
    <title>S. Mantenimientos - GADM ALAUSI</title>
    <body style="text-align: justify;" onMouseOver="hayActividad();">
      <!--<div id="codCas" name="codCas" ><div style="display:none;"><%
/*out.println(codigocas);*/></div></div>-->
      <!--<div id="Logeo" name="Logeo" ><div style="display:none;"><%
/*out.println(tipoLogeo);*/></div></div>-->
      <!-- Nav -->
      <nav id="barraLateral" class="px-nav px-nav-left">
        <button id="dd" type="button" class="px-nav-toggle" data-toggle="px-nav">
          <span class="px-nav-toggle-arrow"></span>
          <span class="navbar-toggle-icon"></span>
          <span class="px-nav-toggle-label font-size-11">ESCONDER MENU</span>
        </button>
```

```

    <ul class="px-nav-content">
        if (objacciones.getAcciones().getAccnombre().equals("Usuarios") &&
usuarios == false) {%>
            <li id="verProveedor" class="px-nav-item">
                <a style="cursor:pointer" onclick="usuarios();" ><i class="fa fa-user-
secret"></i>&nbsp;&nbsp;<span class="px-nav-label">Usuarios</span></a>
            </li>
            <%          usuarios = true;
                }%>

            <%
        </li>
        <li class="px-nav-item px-nav-dropdown">
            <a href="#"><i class="px-nav-icon ion-monitor"></i><span class="px-nav-
label">Equipos de Computo</span></a>
            <ul class="px-nav-dropdown-menu">
                <%
                    for (Accionesrolusuario objacciones : listTipoflujo) {

                        if
(objacciones.getAcciones().getAccnombre().equals("Dependencias") && dependencias
== false) {%>
                            <li id="verProveedor" class="px-nav-item">
                                <a style="cursor:pointer" onclick="dependencias();" ><i class="fa fa-
bathtub"></i>&nbsp;&nbsp;<span class="px-nav-label">Dependencias</span></a>
                            </li>
                            <%          dependencias = true;
                                }

                                if (objacciones.getAcciones().getAccnombre().equals("Equipos   de
Computo") && equipos == false) {%>
                                    <li id="verProveedor" class="px-nav-item">
                                        <a style="cursor:pointer" onclick="equipos();" ><i class="fa fa-arrow-
circle-down"></i>&nbsp;&nbsp;<span          class="px-nav-label">Equipos          de
Computo</span></a>
                                    </li>
                                    <%          equipos = true;
                                        }
                                    %>
                                <%          }%>
                            </ul>
                        </li>
                        <li class="px-nav-item px-nav-dropdown">
                            <a href="#"><i class="px-nav-icon ion-lock-combination"></i><span
class="px-nav-label">Mantenimiento</span></a>
                            <ul class="px-nav-dropdown-menu">
                                <%

```

```

        for (Accionesrolusuario objacciones : listTipoflujo) {

            if
            (objacciones.getAcciones().getAccnombre().equals("Mantenimientos")
            &&
            mantenimientos == false) {%>
                <li id="verProveedor" class="px-nav-item">
                    <a style="cursor:pointer" onclick="mantenimientos();" ><i class="fa
                    fa-archive"></i>&nbsp;&nbsp;&nbsp;<span class="px-nav-
                    label">Mantenimientos</span></a>
                </li>
                <%
                    mantenimientos = true;
                }
            </ul>
        </li>
        <li class="px-nav-item px-nav-dropdown">
            <a href="#"><i class="px-nav-icon ion-ios-chatboxes"></i><span
            class="px-nav-label">Reportes</span></a>
            <ul class="px-nav-dropdown-menu">
                <%
                    for (Accionesrolusuario objacciones : listTipoflujo) {

                        if
                        (objacciones.getAcciones().getAccnombre().equals("Reporte
                        Total Mantenimientos") && reportetotal == false) {%>
                            <li id="verProveedor" class="px-nav-item">
                                <a style="cursor:pointer" onclick="UserPdf();" ><i class="fa fa-
                                bitbucket"></i>&nbsp;&nbsp;&nbsp;<span class="px-nav-label">Total
                                Mantenimientos</span></a>
                            </li>
                        </ul>
                    </nav>
                <!-- Navbar -->
                <nav class="navbar px-navbar">
                    <div class="navbar-header">
                        <a class="navbar-brand" href="index.jsp"><b>Sistema de Mantenimientos de
                        equipos de Computos</b></a>
                    </div>
                if (session.getAttribute("ingreso").toString().equals("true")) {
                    boolean administrador = false;
                    boolean tecnico = false;
                    listroles
                    =
                    lista.listaderolesmenudadodcedula(request.getAttribute("cedula").toString());

                    for (Roles objrol : listroles) {
                        if
                        (objrol.getRoldescripcion().equals("ADMINISTRADOR")
                        && administrador == false) {
                            out.println("<li><a
                            style='cursor:pointer'"
                            +
                            objrol.getRoldescripcion() + "</a></li>");

```

```

        administrador = true;
    }
    if (objrol.getRoldescripcion().equals("TECNICO") && tecnico
== false) {
        out.println("<li><a          style='cursor:pointer'>"          +
objrol.getRoldescripcion() + "</a></li>");
        tecnico = true;
    }
}
%>
<!-- Content -->
<div class="px-content">
    <div id="contenidoDinamico">
        <div class="page-header">
            <h1>El GADM ALAUSÍ le da la cordial bienvenida.</h1>
        </div>
        <div id="alert-dtic"></div>

        <div>
            <h1 class="m-t-0 m-b-1 text-xs-center text-muted font-weight-
normal">Sistema de Mantenimiento de Equipos de Computo</h1>
            </br>
            <div class="panel-img p-a-1">
                
            </div>
            </br>
            <div class="page-wide-block p-a-4 bg-white font-size-15">
            </div>
            </br>
            <div class="row">
                <div class="col-md-6">
                    <div class="panel text-xs-center">
                        <div class="panel-body">
                            <p class="p-y-1"><i class="ion-ios-chatboxes font-size-46 line-
height-1 text-primary"></i></p>
                            <p><strong>Tiene preguntas?</strong></p>
                            <p class="m-b-4">Puede contactarse a nuestro <a
href="http://www.alausi.gob.ec/index.php/post-formats" target="_blank"> servicio de
atención </a> estaremos gustosos de atender todas sus dudas.</p>
                        </div>
                    </div>
                </div>
                <div class="col-md-6">
                    <div class="panel text-xs-center">
                        <div class="panel-body">

```

```

        <p class="p-y-1"><i class="ion-ios-medkit font-size-46 line-
height-1 text-primary"></i></p>
        <p><strong>Soporte</strong></p>
        <p class="m-b-4">Necesita actualizar sus datos o tiene algún
inconveniente en los mismos, por favor comuníquese con nosotros para poder brindarle
una solución rápida.</p>
    </div>
</div>
</div>

<hr class="page-wide-block m-t-0">

<hr class="page-wide-block m-t-0">

</div>
</div>
</div>
<!-- Footer -->
<footer class="px-footer px-footer-bottom">
    Copyright © 2020 GADM ALAUSÍ. Todos los derechos reservados.
</footer>
<link href="css/bootstrap.min.css" rel="stylesheet" type="text/css">
<link href="css/pixeladmin.min.css" rel="stylesheet" type="text/css">
<link href="css/widgets.min.css" rel="stylesheet" type="text/css">

```

## 10. Equipo

Esta codificación es para mostrar todas las propiedades del equipo, así como también mostrará y efectuará las opciones para modificar y eliminar.

**Nombre archivo:** Equipos.jsp

El archivo que contine este código fuente se encuentra en el servidor en la siguiente ruta:

**D:\payara41\domains1\sistemas\matenimientoIU\src\main\webapp\Equipos.jsp**

```

<% @ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link rel="stylesheet" href="content/css/bootstrap.min.css"
crossorigin="anonymous">
<link rel="stylesheet" href="content/css/Main.css" crossorigin="anonymous">
<title>Mantenimiento</title>
</head>
<body>
<div class="d-flex">
<div class="col-sm-4">
<div class="card">
<div class="card-body">
<div class="form-group">
<label>Equipo de computo</label>
</div>
<c:choose>
<c:when test="{{(cond == 0)}}">
<div class="alert alert-danger alert-dismissible">Error</div>
</c:when>
<c:when test="{{(cond == 1)}}">
<div class="alert alert-success alert-dismissible">Creado
 exitosamente.</div>
</c:when>
<c:when test="{{(cond == 2)}}">
<div class="alert alert-success alert-dismissible">Actualizado
 exitosamente.</div>
</c:when>
<c:when test="{{(cond == 3)}}">
<div class="alert alert-success alert-dismissible">Eliminado
 exitosamente.</div>
</c:when>
<c:otherwise>
</c:otherwise>
</c:choose>
<form action="Opciones?menu=Equipo" method="POST">

```

```

        <input type="hidden" value="{ equipo.getIdEquipo()}"
name="txtId" class="form-control">
        <div class="form-group">
            <label>Nombre</label>
            <input type="text" value="{ equipo.getNombreEquipo()}"
name="txtNombre" class="form-control">
        </div>
        <div class="form-group">
            <label>Marca</label>
            <input type="text" value="{ equipo.getNmarcaEquipo()}"
name="txtMarca" class="form-control">
        </div>
        <div class="form-group">
            <label>Serie</label>
            <input type="text" value="{ equipo.getNserieEquipo()}"
name="txtSerie" class="form-control">
        </div>
        <div class="form-group">
            <label>Daño</label>
            <input type="text" value="{ equipo.getNdanioEquipo()}"
name="txtDanio" class="form-control">
        </div>
        <input type="submit" name="accion" value="Agregar" class="btn btn-
primary">
        <input type="submit" name="accion" value="Actualizar" class="btn
btn-success">
    </form>
</div>
</div>
</div>
<div class="col-sm-8">
    <div class="card">
        <div class="card-body">
            <table class="table table-hover">

```



```

<thead>
  <tr>
    <th>Nombre</th>
    <th>Marca</th>
    <th>Serie</th>
    <th>Daño</th>
    <th style="text-align: center">ACCIONES</th>
  </tr>
</thead>
<tbody>
  <c:forEach var="em" items="{equipos}">
    <tr>
      <td>${em.getNnombreEquipo()}</td>
      <td>${em.getNmarcaEquipo()}</td>
      <td>${em.getNserieEquipo()}</td>
      <td>${em.getNdanioEquipo()}</td>
      <td style="text-align: center">
        <a class="btn btn-warning"
href="Opciones?menu=Equipo&accion=Editar&id=${em.getNidEquipo()}">Editar</a>
        <a class="btn btn-danger"
onclick="confirmar('Opciones?menu=Equipo&accion=Delete&id=${em.getNidEquipo(
)})">Delete</a>
      </td>
    </tr>
  </c:forEach>
</tbody>
</table>
</div>
</div>
</div>
</div>
<script src="content/js/jquery-3.3.1.slim.min.js"
crossorigin="anonymous"></script>
<script src="content/js/popper.min.js" crossorigin="anonymous"></script>

```

```
<script src="content/js/bootstrap.min.js" crossorigin="anonymous"></script>
<script src="content/js/funciones.js" crossorigin="anonymous"></script>
</body>
</html>
```

## 11. Usuarios

El siguiente código fuente que se muestra a continuación, es el que nos permitirá mostrar una lista de usuarios ya ingresados, así como también tenemos las opciones de modificar los datos y desactivar, ya que el usuario solo debe tener un eliminado lógico por los históricos que maneja el sistema.

### Nombre archivo: UsuariosJpaController.java

El archivo que contine este código fuente se encuentra en el servidor en la siguiente ruta:

**D:\payara41\domains1\sistemas\matenimientoIU\src\main\java\controllers\UsuariosJpaController.java**

```
package controllers;

import javax.persistence.Query;
import javax.persistence.EntityNotFoundException;
import javax.persistence.criteria.CriteriaQuery;
import javax.persistence.criteria.Root;
import entidadesMa.Accionesrolusuario;
import java.util.ArrayList;
import java.util.Collection;
import entidadesMa.Mantenimiento;
import entidadesMa.Usuarios;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.transaction.UserTransaction;

public class UsuariosJpaController implements Serializable {
    public UsuariosJpaController(UserTransaction utx, EntityManagerFactory emf) {
        this.utx = utx;
        this.emf = emf;
    }
}
```

```

}
private UserTransaction utx = null;
private EntityManagerFactory emf = null;

public EntityManager getEntityManager() {
    return emf.createEntityManager();
}

public void create(Usuarios usuarios) throws RollbackFailureException, Exception {
    if (usuarios.getAccionesrolusuarioCollection() == null) {
        usuarios.setAccionesrolusuarioCollection(new
ArrayList<Accionesrolusuario>());
    }
    if (usuarios.getMantenimientoCollection() == null) {
        usuarios.setMantenimientoCollection(new ArrayList<Mantenimiento>());
    }
    EntityManager em = null;
    try {
        utx.begin();
        em = getEntityManager();
        Collection<Accionesrolusuario> attachedAccionesrolusuarioCollection = new
ArrayList<Accionesrolusuario>();
        for (Accionesrolusuario
accionesrolusuarioCollectionAccionesrolusuarioToAttach :
usuarios.getAccionesrolusuarioCollection()) {
            accionesrolusuarioCollectionAccionesrolusuarioToAttach =
em.getReference(accionesrolusuarioCollectionAccionesrolusuarioToAttach.getClass(),
accionesrolusuarioCollectionAccionesrolusuarioToAttach.getAccionesrolusuarioPK());
            attachedAccionesrolusuarioCollection.add(accionesrolusuarioCollectionAccionesrolusu
arioToAttach);
        }
        usuarios.setAccionesrolusuarioCollection(attachedAccionesrolusuarioCollection);
        Collection<Mantenimiento> attachedMantenimientoCollection = new
ArrayList<Mantenimiento>();

```

```

        for (Mantenimiento mantenimientoCollectionMantenimientoToAttach :
usuarios.getMantenimientoCollection()) {
            mantenimientoCollectionMantenimientoToAttach =
em.getReference(mantenimientoCollectionMantenimientoToAttach.getClass(),
mantenimientoCollectionMantenimientoToAttach.getIdmantenimiento());

attachedMantenimientoCollection.add(mantenimientoCollectionMantenimientoToAttac
h);
        }
        usuarios.setMantenimientoCollection(attachedMantenimientoCollection);
        em.persist(usuarios);
        for (Accionesrolusuario accionesrolusuarioCollectionAccionesrolusuario :
usuarios.getAccionesrolusuarioCollection()) {
            for (Mantenimiento mantenimientoCollectionMantenimiento :
usuarios.getMantenimientoCollection()) {
                Usuarios oldIntusuarioOfMantenimientoCollectionMantenimiento =
mantenimientoCollectionMantenimiento.getIntusuario();
                mantenimientoCollectionMantenimiento.setIntusuario(usuarios);
                mantenimientoCollectionMantenimiento =
em.merge(mantenimientoCollectionMantenimiento);
                if (oldIntusuarioOfMantenimientoCollectionMantenimiento != null) {

oldIntusuarioOfMantenimientoCollectionMantenimiento.getMantenimientoCollection().
remove(mantenimientoCollectionMantenimiento);
                oldIntusuarioOfMantenimientoCollectionMantenimiento =
em.merge(oldIntusuarioOfMantenimientoCollectionMantenimiento);
                }
            }
            utx.commit();
        } catch (Exception ex) {
            try {
                utx.rollback();
            } catch (Exception re) {

```

```

        throw new RollbackFailureException("An error occurred attempting to roll
back the transaction.", re);
    }
    throw ex;
} finally {
    if (em != null) {
        em.close();
    }
}
    utx.commit();
} catch (Exception ex) {
    try {
        utx.rollback();
    } catch (Exception re) {
        throw new RollbackFailureException("An error occurred attempting to roll
back the transaction.", re);
    }
    String msg = ex.getLocalizedMessage();
    if (msg == null || msg.length() == 0) {
        Integer id = usuarios.getIdusuario();
        if (findUsuarios(id) == null) {
            throw new NonexistentEntityException("The usuarios with id " + id + " no
longer exists.");
        }
    }
    throw ex;
} finally {
    if (em != null) {
        em.close();
    }
}
}

public List<Usuarios> findUsuariosEntities() {
    return findUsuariosEntities(true, -1, -1);
}

```

```

    }
    public List<Usuarios> findUsuariosEntities(int maxResults, int firstResult) {
        return findUsuariosEntities(false, maxResults, firstResult);
    }
    private List<Usuarios> findUsuariosEntities(boolean all, int maxResults, int
firstResult) {
        EntityManager em = getEntityManager();
        try {
            CriteriaQuery cq = em.getCriteriaBuilder().createQuery();
            cq.select(cq.from(Usuarios.class));
            Query q = em.createQuery(cq);
            if (!all) {
                q.setMaxResults(maxResults);
                q.setFirstResult(firstResult);
            }
            return q.getResultList();
        } finally {
            em.close();
        }
    }
}

```

```

public Usuarios findUsuarios(Integer id) {
    EntityManager em = getEntityManager();
    try {
        return em.find(Usuarios.class, id);
    } finally {
        em.close();
    }
}

```

```

public int getUsuariosCount() {
    EntityManager em = getEntityManager();
    try {
        CriteriaQuery cq = em.getCriteriaBuilder().createQuery();

```

```

    Root<Usuarios> rt = cq.from(Usuarios.class);
    cq.select(em.getCriteriaBuilder().count(rt));
    Query q = em.createQuery(cq);
    return ((Long) q.getSingleResult()).intValue();
} finally {
    em.close();
}
}
}

```

## 12. Mantenimiento

Es el código fuente esta en lenguaje de programación Java, dicho código permite unir todas las opciones para conformar el mantenimiento, usuarios, equipos responsables fechas etc.

### Nombre archivo: MantenimientoJpaController.java

Este archivo se encuentra en el servidor en la siguiente ruta:

**D:\payara41\domains1\sistemas\matenimientoIU\src\main\java\controllers\MantenimientoJpaController.java**

```

package controllers;
import controllers.exceptions.NonexistentEntityException;
import controllers.exceptions.RollbackFailureException;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.transaction.UserTransaction;
public class MantenimientoJpaController implements Serializable {
    public MantenimientoJpaController(UserTransaction utx, EntityManagerFactory emf) {
        this.utx = utx;
        this.emf = emf;
    }
    private UserTransaction utx = null;
    private EntityManagerFactory emf = null;

```

```

public void create(Mantenimiento mantenimiento) throws RollbackFailureException,
Exception {
    EntityManager em = null;
    try {
        utx.begin();
        em = getEntityManager();
        Usuarios intusuario = mantenimiento.getIntusuario();
        if (intusuario != null) {
            intusuario = em.getReference(intusuario.getClass(),
intusuario.getIdusuario());
            mantenimiento.setIntusuario(intusuario);
        }
        em.persist(mantenimiento);
        if (intusuario != null) {
            intusuario.getMantenimientoCollection().add(mantenimiento);
            intusuario = em.merge(intusuario);
        }
        utx.commit();
    } catch (Exception ex) {
        try {
            utx.rollback();
        } catch (Exception re) {
            throw new RollbackFailureException("An error occurred attempting to roll
back the transaction.", re);
        }
        throw ex;
    } finally {
        if (em != null) {
            em.close();
        }
    }
}

```



```

public void edit(Mantenimiento mantenimiento) throws NonexistentEntityException,
RollbackFailureException, Exception {
    EntityManager em = null;
    try {
        utx.begin();
        em = getEntityManager();
        Mantenimiento persistentMantenimiento = em.find(Mantenimiento.class,
mantenimiento.getIdmantenimiento());
        Usuarios intusuarioOld = persistentMantenimiento.getIntusuario();
        Usuarios intusuarioNew = mantenimiento.getIntusuario();
        if (intusuarioNew != null) {
            intusuarioNew = em.getReference(intusuarioNew.getClass(),
intusuarioNew.getIdusuario());
            mantenimiento.setIntusuario(intusuarioNew);
        }
        mantenimiento = em.merge(mantenimiento);
        if (intusuarioOld != null && !intusuarioOld.equals(intusuarioNew)) {
            intusuarioOld.getMantenimientoCollection().remove(mantenimiento);
            intusuarioOld = em.merge(intusuarioOld);
        }
        if (intusuarioNew != null && !intusuarioNew.equals(intusuarioOld)) {
            intusuarioNew.getMantenimientoCollection().add(mantenimiento);
            intusuarioNew = em.merge(intusuarioNew);
        }
        utx.commit();
    } catch (Exception ex) {
        try {
            utx.rollback();
        } catch (Exception re) {
            throw new RollbackFailureException("An error occurred attempting to roll
back the transaction.", re);
        }
        String msg = ex.getLocalizedMessage();
        if (msg == null || msg.length() == 0) {

```

```

        Integer id = mantenimiento.getIdmantenimiento();
        if (findMantenimiento(id) == null) {
            throw new NonexistentEntityException("The mantenimiento with id " + id
+ " no longer exists.");
        }
    }
    throw ex;
} finally {
    if (em != null) {
        em.close();
    }
}
}

public List<Mantenimiento> listaderolesmenudadodcedula(String fechainicio, String
fechafin) {
    EntityManager em = getEntityManager();
    List<Mantenimiento> objpersona = null;
    String consulta;
    try {
        consulta = "SELECT m FROM Mantenimiento m WHERE m.fechaIngreso >= "
+ fechainicio + " AND m.fechaIngreso <=" + fechafin + """;
        Query query = em.createQuery(consulta);
        List<Mantenimiento> listainstitucion = query.getResultList();
        objpersona = listainstitucion;
    } catch (Exception e) {
        System.out.println("dd" + e.toString());
    }
    return objpersona;
}

public List<Mantenimiento> listademantenimientosdadoequipo(Integer equipo) {
    EntityManager em = getEntityManager();
    List<Mantenimiento> objpersona = null;
    String consulta;
    try {

```

```

        consulta = "SELECT m FROM Mantenimiento m WHERE
m.intequipo.idequipo =?1";
        Query query = em.createQuery(consulta);
        query.setParameter(1, equipo);
        List<Mantenimiento> listainstitucion = query.getResultList();
        objpersona = listainstitucion;
    } catch (Exception e) {
        System.out.println("dd" + e.toString());
    }
    return objpersona;
}
}

```

### 13. Reportes

Este Código Fuente permite mostrar y generar los reports tanto visualmente como poder descargar en formato PDF.

#### Nombre archivo: UserPdf.jsp

La ubicación para su despliegue en el servidor se encuentra en la siguiente ruta:

**D:\paya41\domains1\sistemas\matenimientoIU\src\main\webapp\UserPdf.jsp**

```

<% @page import="java.util.Base64"%>
<% @page import="java.io.IOException"%>
<% @page import="java.io.FileInputStream"%>
<% @page contentType='text/html' pageEncoding='UTF-8'%>
<!DOCTYPE html>
<html>
    <script src="assets/js/jquery/jquery-2.1.0.min.js"></script>
    <script src="assets/js/plugins/datatables/jquery.dataTables.min.js"></script>
    <script src="assets/js/plugins/datatables/exts/dataTables.colVis.bootstrap.js"></script>
    <script src="assets/js/king-table.js"></script>
    <style>.video-responsive {
        position: relative;
        padding-bottom: 56.25%; /* 16/9 ratio */
    }

```

```

padding-top: 20px; /* IE6 workaround*/
}
</style>
<script src="js/jquery.loadingModal.min.js" type="text/javascript"></script>
<link href="js/css/jquery.loadingModal.min.css" rel="stylesheet" type="text/css"/>
<%
String codUser = "dd";
String encodedBase64 = "";
String sSistemaOperativo = System.getProperty("os.name");
String string = sSistemaOperativo;
String[] parts = string.split(" ");
String part1 = parts[0]; // 123
GeneratePDFFileIText generatePDFFileIText = new GeneratePDFFileIText();
String aaa = GeneratePDFFileIText.RutasComprobantes(part1);
String ruta = GeneratePDFFileIText.TotalMantenimientos(new File(aaa +
"reporte" + codUser + ".pdf"));
codUser = codUser.trim();
File archivo = new File(aaa + "reporte" + codUser + ".pdf");
boolean estatus = false;
FileInputStream fileInputStreamReader = new FileInputStream(archivo);
byte[] bytes = new byte[(int) archivo.length()];
fileInputStreamReader.read(bytes);
fileInputStreamReader.close();
encodedBase64 = new String(Base64.getEncoder().encode(bytes));
estatus = archivo.delete();
if (!estatus) {
    response.sendRedirect("index.jsp");
}
%>
<div class="main-header">
<h2>Mantenimientos de Equipos de Computo</h2>
<em>Se encontrará en formato PDF toda la información ingresada.</em>
</br> </div>
<div id='contenidoInferior' class="main-content">

```

