

UNIVERSIDAD TECNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS COMPUTACIONALES

TEMA:

**SISTEMA DE EVALUACIÓN, CONTROL Y REPORTES PARA EL
LABORATORIO DE EMAPA-I, PARROQUIA DE CARANQUI
UTILIZANDO HERRAMIENTAS LIBRES.**

APLICATIVO:

**CONTROL DE LAS MUESTRAS DE AGUA GENERADAS EN EL
LABORATORIO DE EMAPA-I, DE LA CIUDAD DE IBARRA, PARROQUIA
DE CARANQUÍ, UTILIZANDO HERRAMIENTAS LIBRES.**

AUTOR:

Gabriela Patricia Jácome Quelal

DIRECTOR:

Ing. Diego Ortiz

Ibarra – Ecuador

2011

CERTIFICACIÓN

Por medio del presente certifico:

Que la estudiante de la Facultad en Ciencias Aplicadas de la Escuela de Ingeniería en Sistemas Computacionales de la Universidad Técnica del Norte, Srta. Gabriela Patricia Jácome Quelal CC 1040122130-4, es autor intelectual y materiales del Proyecto de Grado con el tema “Sistema de Evaluación, Control y Reportes para el Laboratorio de EMAPA-I, parroquia de Caranqui utilizando herramientas libres.”, está certificación la confiero por haber desempeñado las funciones de Director del Proyecto durante todo el tiempo que se ocupó en la elaboración y desarrollo del mencionado proyecto.

Ing. Diego Ortiz

DIRECTOR DE TESIS

DEDICATORIA

Este trabajo va dedicado:

A Dios, por permitirme llegar a este momento tan especial en mi vida. Por los triunfos y los momentos difíciles que me han enseñado a valorar lo especial de la vida día a día.

A todas aquellas personas quienes buscan usar la ciencia en pos de ayudar a los demás.

A mi familia en especial a mi hijo Leandro David que me dio la inspiración y las fuerzas necesarias para poder culminar una de mis metas.

A mis padres que me dieron la vida y que gracias a ellos he podido llegar a cumplir una de mis metas, en especial a mi madre por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación firme que me ha permitido ser una persona de bien, pero más que nada, por su amor y su comprensión.

AGRADECIMIENTO

A Dios, por darme las fuerzas necesarias para poder culminar mi carrera profesional y darme la oportunidad de tenerlo siempre en mi corazón.

A la Universidad Técnica del Norte y en especial a la Facultad de Ingeniería en Ciencias Aplicadas por las facilidades brindadas para realizar el presente trabajo.

Al personal docente de la Escuela de Ingeniería en Sistemas Computacionales por su incansable labor en beneficio de la juventud estudiosa del norte del país.

Al ingeniero Diego Ortiz, que como docente supo impartir sus sólidos conocimientos para mi carrera, motivándome siempre a la investigación y como Director del Proyecto incentivándome a la culminación del mismo .

A la EMAPA-I por facilitar sus instalaciones para la implantación del aplicativo del proyecto de manera especial al Lic. Carlos Hidrobo por prestar su apoyo a la realización del proyecto como jefe de Recursos Informáticos de la empresa.

CONTENIDO.

| | |
|--------------------------------------------------------------|----|
| 1. INTRODUCCION..... | 4 |
| 2. ALCANCE..... | 5 |
| 3. HERRAMIENTAS DE DESARROLLO | 5 |
| 3.1 Apache Tomcat 6.0..... | 6 |
| 3.2 Lenguaje de Programación JSP con IDE NetBeans 5 .5 | 9 |
| 3.2.1 Ventajas..... | 10 |
| 3.2.2 Ciclos de Vida del Servlet..... | 11 |
| 3.2.3 Consecuencias del Ciclo de Vida del Servlet..... | 13 |
| 3.2.4 NetBeans 5.5 | 14 |
| 3.3 Motor de Base de Datos PostgreSQL v 8.4..... | 15 |
| 3.3.1 Breve Historia | 16 |
| 3.3.2 Prestaciones..... | 17 |
| 3.3.3 Limitaciones..... | 22 |
| 4. BENEFICIOS DEL SISTEMA | 22 |
| 1. VISIÓN..... | 25 |
| 1.1 Propósito..... | 25 |
| 1.2 Alcance..... | 25 |
| 1.3 Posicionamiento | 26 |
| 1.4 Descripción de los Interesados y Usuarios..... | 29 |
| 1.5 Perfiles de los Stakeholders..... | 32 |
| 1.6 Vista General del Producto..... | 38 |
| 1.7 Costos y Precios | 40 |

| | | |
|--------|--------------------------------------------------|----|
| 1.8 | Características del Producto | 41 |
| 1.9 | Restricciones | 42 |
| 1.10 | Programación por Capas | 43 |
| 1.10.1 | Introducción | 44 |
| 1.10.2 | Características de la Programación en Capas..... | 44 |
| 1.10.3 | Capa de Presentación o Frontera..... | 45 |
| 1.10.4 | Capa de Lógica de Negocio o Control | 46 |
| 1.10.5 | Capa de Datos | 46 |
| 1.10.6 | Ventajas y Desventajas | 47 |
| 2. | PLAN DE DESARROLLO DE SOFTWARE | 50 |
| 2.1 | Propósito..... | 50 |
| 2.2 | Alcance | 51 |
| 2.3 | Vista General del Proyecto..... | 51 |
| 2.4 | Organización del Proyecto | 56 |
| 2.5 | Plan del Proyecto..... | 59 |
| 3. | TECNOLOGIA JSP..... | 66 |
| 3.1 | Introducción..... | 66 |
| 3.2 | Antecedentes. | 66 |
| 3.3 | Características. | 67 |
| 3.4 | JSP | 70 |
| 3.4.1 | Motor JSP..... | 75 |
| 3.4.2 | Arquitectura..... | 76 |
| 3.4.3 | Funcionamiento de JSP..... | 78 |

| | | |
|-------|------------------------------------------------------------------------|-----|
| 3.4.4 | Componentes de una página JSP..... | 80 |
| 3.4.5 | JAVABEANS | 84 |
| 3.4.6 | Etiquetas personalizadas (TAG) | 84 |
| 3.4.7 | Servlets | 85 |
| 3.4.8 | Características de los Servlets..... | 86 |
| 3.5 | Acceso a Base de Datos desde JSP. | 87 |
| 3.5.1 | Conectividad a base de datos | 87 |
| 3.5.2 | JDBC..... | 89 |
| 3.5.3 | Drives o controladores JDBC. | 89 |
| 3.6 | SEGURIDAD EN JSP..... | 91 |
| 3.6.1 | Modelos de Seguridad en JAVA..... | 91 |
| 3.6.1 | Dominios protegidos, modelo de permisos y políticas de seguridad. | 92 |
| 3.6.2 | Control de acceso a paginas JSP. | 93 |
| 4. | ESPECIFICACIONES..... | 95 |
| 4.1 | Especificación de casos de Uso..... | 95 |
| 4.1.1 | Ingreso Personal Administrativo..... | 95 |
| 4.1.2 | Ingreso personal Usuarios | 97 |
| 4.1.3 | Ingreso de tipo de variable | 98 |
| 4.1.4 | Ingreso Sectores | 100 |
| 4.1.5 | Ingreso Parámetros Constantes | 102 |
| 4.1.6 | Ingreso Determinación de Cloro Residual. | 103 |
| 4.1.7 | Ingreso Tipo de Parámetros. | 105 |
| 4.1.8 | Ingreso Tipo de Unidades. | 106 |

| | | |
|-------|--------------------------------------------------------------------------------------------------------|-----|
| 4.1.9 | Ingreso de Análisis..... | 107 |
| 4.2 | Diagrama de Actividades | 110 |
| 4.2.1 | Sistema en General..... | 110 |
| 4.2.2 | Ingreso Tipo Variable | 111 |
| 4.2.3 | Ingreso Sectores | 112 |
| 4.2.4 | Parámetros Constantes | 113 |
| 4.2.5 | Tipos Parámetros..... | 114 |
| 4.2.6 | Ingreso de Unidades..... | 115 |
| 4.3 | Diagrama de Despliegue | 115 |
| 4.4 | Diagrama de Secuencia | 116 |
| 4.4.1 | DIAGRAMA DE SECUENCIA: Datos iniciales. Proceso de administración..... | 118 |
| 4.4.2 | DIAGRAMA DE SECUENCIA: Proceso de Generación de Reportes y Proceso de Reportes de Administración | 119 |
| 5. | DOCUMENTACION TECNICA..... | 121 |
| 5.1 | Funcionalidad del Sistema para el Laboratorio de Caranqui | 121 |
| 5.1.1 | Datos Iniciales, Administración y Reportes..... | 123 |
| 5.1.2 | Tareas del Administrador..... | 127 |
| 6. | CONCLUSIONES Y RECOMENDACIONES | 130 |
| 6.1 | CONCLUSIONES..... | 130 |
| 6.2 | RECOMENDACIONES..... | 131 |
| 7. | GLOSARIO | 134 |
| 8. | REFERENCIAS BIBLIOGRAFICAS | 139 |
| 8.1 | REFERENCIAS LIBROS | 139 |

| | | |
|-----|--------------------------------------|-----|
| 8.2 | REFERENCIAS WEB | 140 |
| 9. | ANEXOS | 142 |
| 9.1 | ANEXO A : BASE DE DATOS | 142 |
| 9.2 | ANEXO B: DICCIONARIO DE DATOS | 143 |
| 9.3 | ANEXO C: MANUAL DE INSTALACIÓN | 154 |
| 9.4 | ANEXO D: MANUAL DE USUARIO | 160 |

INDICE DE FIGURAS

| | |
|---------------------------------------------------------------------------------|----|
| Figura 1. Esquema del Proyecto..... | 5 |
| Figura 2. Estructura del Contenedor Tomcat..... | 7 |
| Figura 3: Funcionamiento del Servidor Apache Tomcat..... | 8 |
| Figura 4: Arquitectura de una Aplicación basada en Servlets Puros + JDBC..... | 9 |
| Figura 5: Arquitectura Cliente-Servidor..... | 10 |
| Figura 6: Ciclo de Vida de un Software..... | 12 |
| Figura 7: Funcionamiento de la Clase Extende HttpServlet..... | 13 |
| Figura 8: Arquitectura Java..... | 14 |
| Figura 9: Arquitectura de PostgreSQL y los Procesos de Información General..... | 16 |
| Figura 10: Interacción con los Componentes de un Sistema..... | 17 |
| Figura 11: Arquitectura de PostgreSQL..... | 21 |
| Figura 12: Perspectiva del Software del Laboratorio..... | 39 |
| Figura 13. Programación por Capas..... | 43 |
| Figura 14: Fases de un Proyecto..... | 62 |
| Figura 15: Tecnología JSP..... | 66 |
| Figura16: Estructura de un Proyecto Java..... | 70 |
| Figura17: Compilación del Programa en Diferentes Tiempos..... | 71 |
| Figura18: Funcionamiento de HTML..... | 72 |

| | |
|--------------------------------------------------------------------|-----|
| Figura19: Funcionamiento de XML..... | 73 |
| Figura 20: Modelo de Arquitectura JSP..... | 74 |
| Figura 21: Motor de JSP..... | 75 |
| Figura 22: Esquema de Conversión de una Página en un Servlets..... | 78 |
| Figura 23: Descripción del Funcionamiento de una Página JSP..... | 78 |
| Figura 24: Funcionamiento del Servlets..... | 85 |
| Figura 25: Acceso a la Base de Datos desde JSP..... | 87 |
| Figura 26: Ingreso Personal Administrativo..... | 95 |
| Figura 27: Ingreso Personal Usuarios..... | 97 |
| Figura 28: Ingreso de Tipo Variable..... | 98 |
| Figura 29: Ingreso Sectores..... | 100 |
| Figura 30: Ingreso Parámetros Constantes..... | 102 |
| Figura 31: Ingreso Determinación de Cloro Residual..... | 103 |
| Figura 32: Ingreso Tipo de Parámetros..... | 105 |
| Figura 33: Ingreso Tipo Unidades..... | 106 |
| Figura 34: Ingreso de Análisis..... | 108 |
| Figura 35: Diagrama de Actividaes - Sistema en General..... | 110 |
| Figura 36: Diagrama de Actividades – Ingreso Tipo Variable..... | 111 |
| Figura 37: Diagrama de Actividades – Ingreso Sectores..... | 112 |
| Figura 38: Diagrama de Actividades – Parámetros Constantes..... | 113 |

| | |
|--------------------------------------------------------------------------------------------------------------------|-----|
| Figura 39: Diagrama de Actividades – Tipos Parámetros..... | 114 |
| Figura 40: Diagrama de Actividades – Unidades..... | 115 |
| Figura 41: Diagrama de Despliegue..... | 116 |
| Figura 42: Objeto..... | 117 |
| Figura 43: Evento..... | 117 |
| Figura 44: Diagrama de Secuencia – Datos Iniciales Proceso de Administración..... | 118 |
| Figura 45: Diagrama de Secuencia – Proceso de Generación de Reportes y Procesos de Reportes de Administración..... | 119 |
| Figura 46: Funcionalidad del Sistema de Laboratorio..... | 122 |
| Figura 47: Ingreso de Datos Iniciales en el Sistema..... | 123 |
| Figura 48: Menu Principal del Sistema..... | 124 |
| Figura 49: Base de Datos del Sistema..... | 142 |
| Figura 50: Ingreso al Paquete de Instalación..... | 154 |
| Figura 51: Mensaje de Preparación de la Instalación..... | 154 |
| Figura 52: Selección de Idioma de Instalación..... | 155 |
| Figura 53: Bienvenida a la Instalación de Programa..... | 155 |
| Figura 54: Opciones de Instalación..... | 156 |
| Figura 55: Inicialización del Cluster..... | 157 |
| Figura 56: Habilitar el Lenguaje de Procedimiento..... | 157 |
| Figura 57: Habilitación de Módulos Contrib..... | 158 |

| | |
|-----------------------------------------------------------|-----|
| Figura 58: Proceso de Finalización..... | 159 |
| Figura 59: Estructura de Inicialización del Sistema..... | 162 |
| Figura 60: Menú Principal de Sistema..... | 162 |
| Figura 61: Administración del Catalogo de Variables..... | 163 |
| Figura 62: Tabla para Ingreso de Datos de Variables..... | 164 |
| Figura 63: Ingreso de Datos de Las Variables..... | 164 |
| Figura 64: Administración del Sectores..... | 165 |
| Figura 65: Ingreso de Datos de los Sectores..... | 166 |
| Figura 66: Actualización de Datos de Sectores..... | 166 |
| Figura 67: Administración de Parámetros..... | 167 |
| Figura 68: Ingreso de Parámetros de Variables..... | 168 |
| Figura 69: Datos de los Parámetros de las Variables..... | 168 |
| Figura 70: Administrador de Cloro Residual..... | 169 |
| Figura 71: Determinación de Cloro Residual – Ingreso..... | 170 |
| Figura 72: Ingreso Reportes de Cloro Residual..... | 170 |
| Figura 73: Administración de Tipos de Parámetros..... | 171 |
| Figura 74: Ingreso de Datos de Tipos de Parámetros..... | 172 |
| Figura 75: Actualización de Tipos de Parámetros..... | 172 |
| Figura 76: Administración de Unidades..... | 173 |
| Figura 77: Ingreso de Datos de las Unidades..... | 173 |

| | |
|---------------------------------------------------------|-----|
| Figura 78: Actualización de Unidades..... | 174 |
| Figura 79: Modificación de Sectores..... | 175 |
| Figura 80 :Ingreso de Parametros..... | 175 |
| Figura 81 :Modificasiónn de Sectores..... | 176 |
| Figura 82: Tabla de Parámetros..... | 176 |
| Figura 83: Ingreso de Datos de Tomas Domiciliaras..... | 177 |
| Figura 84: Ingreso de Parámetros..... | 178 |
| Figura 85: Reportes de Datos de Tomas Domiciliaras..... | 178 |
| Figura 86: Tabla de Análisis de Tomas Domiciliaras..... | 179 |

INDICE DE TABLAS

| | |
|-------------------------------------------------------------------|-----|
| Tabla 1.1: Operadores soportados SQL 92 Y SQL 99..... | 21 |
| Tabla 1.2: Definición del Problema..... | 27 |
| Tabla 1.3: Declaración del Posicionamiento del Proyecto..... | 28 |
| Tabla 1.4: Descripción de los Interesados del Proyecto..... | 30 |
| Tabla 1.5: Coordinador del Proyecto..... | 33 |
| Tabla 1.6: Responsable del Proyecto..... | 33 |
| Tabla 1.7: Responsable Funcional..... | 32 |
| Tabla 1.8: Perfiles de los Usuarios..... | 35 |
| Tabla 1.9: Perfiles de los Usuarios..... | 36 |
| Tabla 1.10: Necesidades de los Interesados y de los Usuarios..... | 37 |
| Tabla 1.11: Resumen de Capacidades..... | 40 |
| Tabla 1.12: Costos y Precios..... | 41 |
| Tabla 2.1: Perfiles de los Usuarios..... | 58 |
| Tabla 2.2: Plan de Fases..... | 59 |
| Tabla 2.3: Descripción de las Fases | 61 |
| Tabla 2.4: Gestión de Riesgos..... | 64 |
| Tabla 3.1: Directivas JSP..... | 81 |
| Tabla 3.2: Secuencia de Comandos..... | 82 |
| Tabla A.1: Tabla de Análisis..... | 143 |

| | |
|-------------------------------------------------|-----|
| Tabla A.2: Tabla de Constantes..... | 144 |
| Tabla A.3: Tabla de Determinación de Cloro..... | 145 |
| Tabla A.4: Tabla de Parámetros..... | 146 |
| Tabla A.5: Tabla de Parámetros Contantes..... | 147 |
| Tabla A.6: Tabla de Persona..... | 148 |
| Tabla A.7: Tabla de Registros..... | 149 |
| Tabla A.8: Tabla de Sectores..... | 150 |
| Tabla A.9: Tabla de Tipo de Parámetros..... | 151 |
| Tabla A.10: Tabla de Tipo de Unidades..... | 152 |
| Tabla A.11: Tabla Usuarios..... | 153 |

RESUMEN

El Sistema de Evaluación, Control y Reportes para el Laboratorio de EMAPA-I, parroquia de Caranqui utilizando Herramientas Libres, es una solución a todos los requerimientos y procesos de comunicación, transferencia e integración de la información, que tiene como objetivos básicos, gestionar los volúmenes de información, con rapidez, exactitud y a la vez la generación de reportes, para facilitar la toma de decisiones y mejorar el control del laboratorio, en los ámbitos de análisis químicos, análisis microbiológicos, análisis físicos con sus respectivos catálogos de parámetros y sobre todo la administración del Laboratorio .

En este objetivo son de gran ayuda la comodidad, confidencialidad, la filosofía abierta, y el conectar capacidades con conocimientos y tecnología, de esta manera El Sistema de Evaluación, Control y Reportes para el Laboratorio de EMAPA-I, parroquia de Caranqui utilizando Herramientas Libres, se ha implementado y puesto en ejecución para contribuir en el desarrollo y productividad de la empresa de agua potable EMAPA-I.

SUMMARY

Evaluation System, Control and Laboratory Reports for EMAPA-I, parish Caranqui using free resources, is a solution to all requirements and communication processes, transfer and integration of information, whose basic objectives, manage volumes of information quickly, accurately and also generate reports to facilitate decision making and better control of the laboratory, in the fields of chemical analysis, microbiological tests, physical tests with their respective catalogs of parameters and especially Laboratory management.

This objective is of great help comfort, privacy, open philosophy, and connecting with knowledge and technology capabilities, thus Evaluation System, Control and Laboratory Reports for EMAPA-I, parish Caranqui using Free resources has been implemented and implemented to assist in the development and productivity of the water utility EMAPA-I.

Laboratorio de
Caranqui

INTRODUCCIÓN



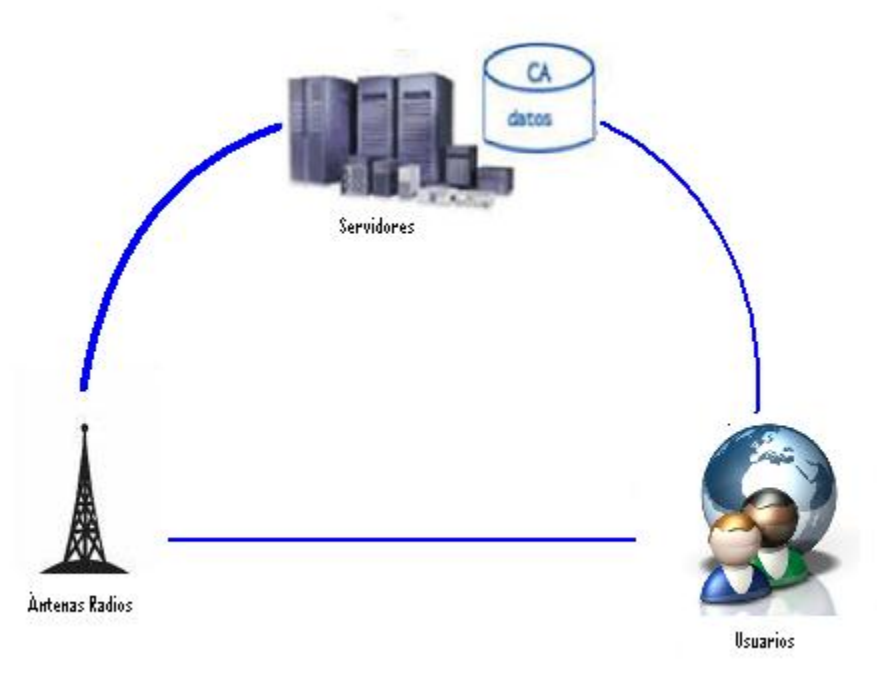
1. INTRODUCCION

En la actualidad todo se maneja en base a sistemas informáticos, debido a los avances en el desarrollo de las tecnologías, de las telecomunicaciones y de la informática. Para toda institución esta dinámica plantea la necesidad de ser transformadas de manera inmediata. En la Empresa de EMAPA-I ubicada en la ciudad de Ibarra los sistemas de Información tienen puntos vulnerables, debido a la fragilidad de la información que se utilizaba, sin la adecuada responsabilidad, con redundancia e inconsistencias; lo que plantea una verdadera dinámica de cambio hacia una efectiva transformación de sus procesos que genere ventaja competitiva a la Institución .

Por ende la responsabilidad de la empresa EMAPA-I es administrar el sistema del Laboratorio de Caranqui de manera integral; para determinar la evaluación, el control y la generación de reportes, garantizando que el mismo sea progresivo; mediante una gestión coordinada y eficiente a fin de que el mismo sea de uso multifunción, orientado al fortalecimiento de la empresa, a la vez, integrarse a un solo repositorio de datos aportando así a una nueva generación, utilizando las potencialidades del JAVA SERVICES PAGES(JSP) y de la Base de Datos PostgreSQL. Este Sistema de evaluación, control y reportes del Laboratorio de Caranqui de la empresa EMAPA-I, está orientado a la eficacia y efectividad con el resultado de datos o reportes a tiempo real y según la necesidad de todos los usuarios y todas las autoridades que integran la empresa. Es así, que con la implantación de este sistema se desea conseguir un buen rendimiento y funcionalidad del Laboratorio de Caranqui, y convertirse en una entidad que busca el éxito a partir de nuevas filosofías con nuevos enfoques, que se conforman a partir del software, con funciones actualizadas y cuyo rendimiento sea permanente.

2. ALCANCE

Desarrollar e implementar una aplicación informática institucional que solucione los problemas de evaluación y control del Laboratorio de Caranqui en la empresa EMAPA-I, apegado a los requerimientos de la empresa.



Fuente: Propia

Figura 1. Esquema del Proyecto

3. HERRAMIENTAS DE DESARROLLO

Para el desarrollo de este proyecto se utilizó:

- 3.1. Servidor de aplicaciones Apache Tomcat 6.0
- 3.2. Lenguaje de programación JSP con IDE NetBeans 5 .5

3.3. Motor de base de datos PostgreSQL v 8.4

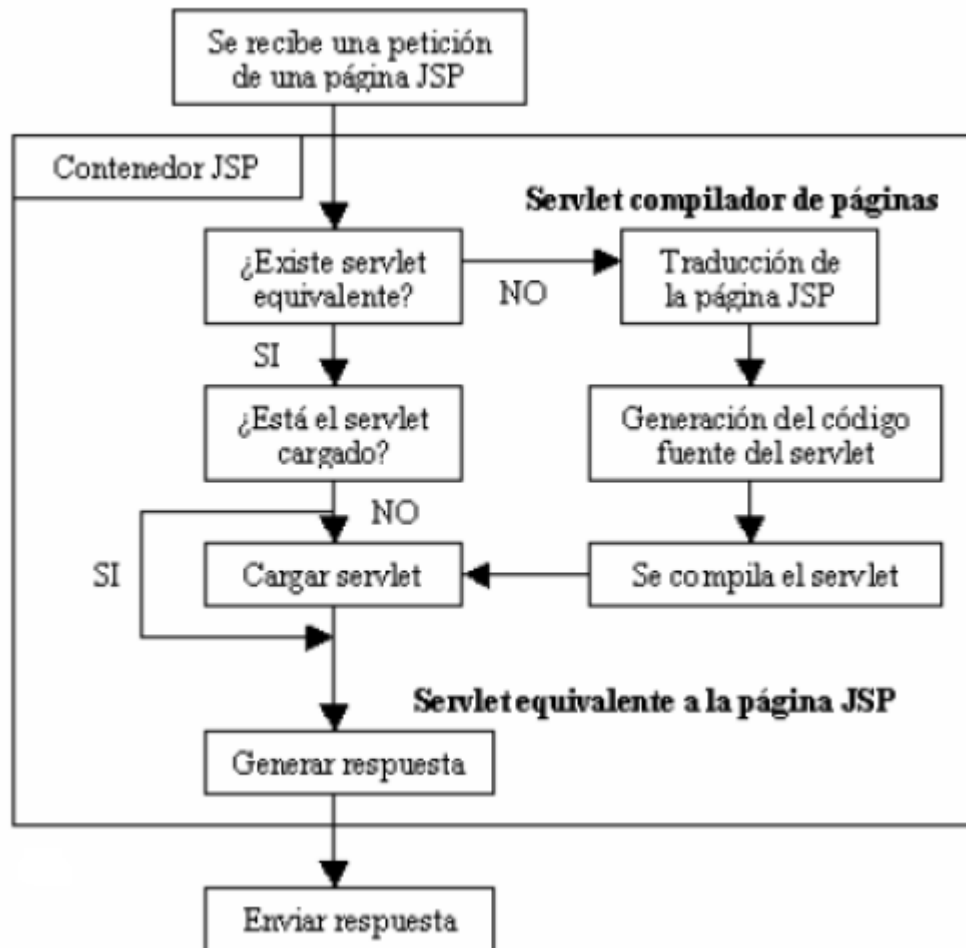
Todas estas herramientas son de licencia *Open Source*¹, lo que resulta en una gran ventaja para que pueda ser asequible a las *Pymes*² (Pequeñas y Medianas Empresas).

3.1 Apache Tomcat 6.0

“El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual. “ Vivek Chopra, Anaya Multimedia-2008. Porque el servidor Apache Tomcat es Open Source y de fácil modificación y su funcionalidad trabaja en diferentes plataformas.

¹ **Open Source.** Es el software estudiar, modificar y mejorar su diseño mediante la disponibilidad de su código fuente. que está licenciado de tal manera que los usuarios pueden

² **Pymes.** Pequeñas y medianas empresas.



Fuente: Castillo Gonzales-Profesores

Figura 2. Estructura del Contenedor Tomcat

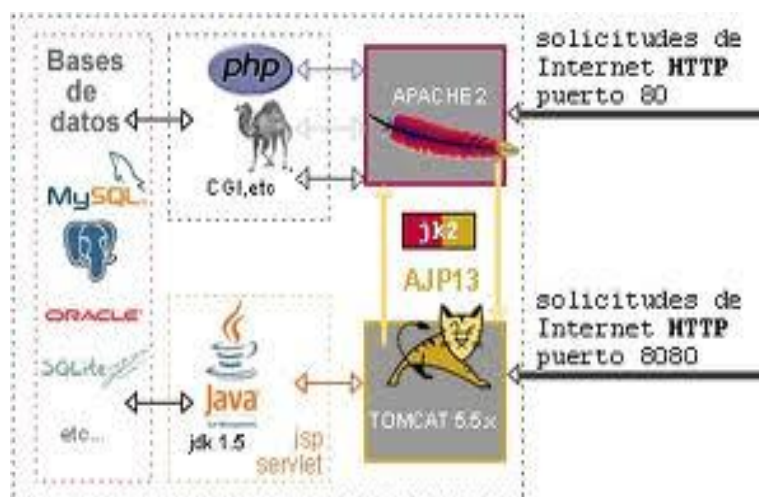
Cuando comenzó su desarrollo en el año 1995, se basó inicialmente en código del popular NCSA HTTPd, pero más tarde fue modificado por completo. Su nombre se debe a que Behelendorf quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde el año 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en el año 2005, siendo el servidor más empleado, en aproximadamente el 70% de los sitios web del mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. (Estadísticas históricas y de uso diario proporcionadas por Netcraft).

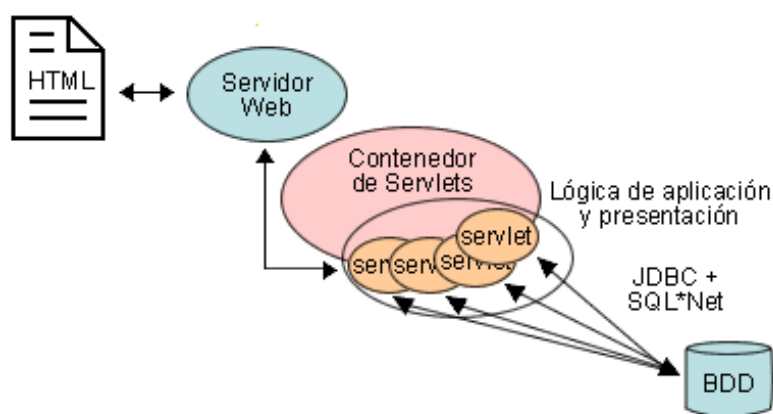
La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales maliciosos en las disposiciones de recibimiento compartidas que utilizan JSP como módulo de Apache.



Fuente: laneros.

Figura 3: Funcionamiento del Servidor Apache Tomcat

3.2 Lenguaje de Programación JSP con IDE NetBeans 5.5

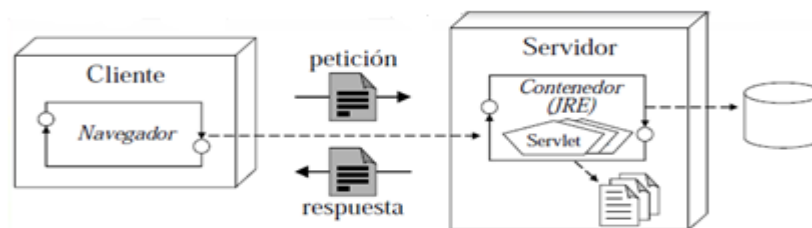


Fuente: greeneyed.org

Figura 4: Arquitectura de una Aplicación basada en Servlets Puros + JDBC

“Un *servlet*³ es una clase java usada para extender las capacidades de los servidores que albergan aplicaciones accedidas mediante un modelo de programación cliente-servidor. Se usa para extender las capacidades de la web comparándose con un programa CGI (Common Gateway Interface) pero con una arquitectura de ejecución diferente, además gestionados por un contenedor de servlets o un motor JVM + implementación del API del servlet” Juan Antonio Palos, 2004. Un servlet es un objeto que corre dentro del contexto de un contenedor y esto expande su funcionalidad.

³ **Servlet.** Aplicación sin interfaz gráfica que se ejecuta en un servidor de Internet, procesando información HTML previamente recogida por un navegador web.



Fuente: Web Component Development With Servlet and JSP Technologies
Sun Microsystems (course SL-314-EE5)

Figura 5: Arquitectura Cliente-Servidor

Dentro del tema servlet se tienen que implementar el interfaz **Servlet**, que define los métodos de ciclo de vida, o bien se tiene que heredar de las siguientes clases:

GenericServlet: esta clase proporciona la interfaz para que el servidor le pase las peticiones al servlet y el mecanismo para controlar el ciclo de vida del servlet.

HttpServlet: para manejar servicios HTTP específicos permitiendo la transferencia de archivos principalmente, en formato HTML. Entre un navegador (el cliente) y un servidor web

ExtendsGenericServlet. Esta clase se hereda o se extiende de la clase `GenericServlet`

3.2.1 Ventajas

Al utilizar servlet nosotros dispondremos de algunas ventajas como pueden ser:

✓ Eficiencia

En un sistema que maneja servlets la eficiencia será medida a través de hilos por cada petición para una única instancia de cada servlet. Las ventajas principales, que entre las cuales tenemos: el rendimiento, es decir, que no hay retraso en las peticiones, las

espaciales que son las de menor consumo de energía y de escalabilidad es la que extiende el margen de operaciones sin perder la calidad de los servicios ofrecidos.

Dentro de las diferentes peticiones que realiza un servlet como pueden ser la conexión a la base de datos, la conexión de red, siempre mantienen su estado cuando realiza las diferentes invocaciones a través de la ejecución de peticiones mediante la invocación de los métodos Get y Post.

Las Utilidades que muestran los servlet para realizar las típicas tareas de servidor, tales como *logging*⁴, gestión de errores, *cookies*⁵, *sesiones*⁶.

✓ Comunicación

La manera que se comunica un servlet con el servidor es de manera estandarizada, comparte datos y crea pools para acceder a la base de datos.

3.2.2 Ciclos de Vida del Servlet

Ciclo de vida de Software.

El término de ciclo de vida del software representa el desarrollo de software, desde la creación hasta la finalización. La intención de este programa es precisar las diferentes fases intermedias que se requiere para validar el desarrollo de la aplicación, es decir, para que el software se garantizado tiene que cumplir los requisitos para la aplicación y comprobación de los pasos de desarrollo

⁴ **Logging.** El registro de algo (Usuario)

⁵ **Cookies.** Son pequeños archivos de texto que son descargados automáticamente, al navegador en una página Web específica.

⁶ **Secciones.** Es la secuencia de páginas que un usuario visita en un sitio web.



Fuente: Mariale440

Figura 6: Ciclo de vida de un software

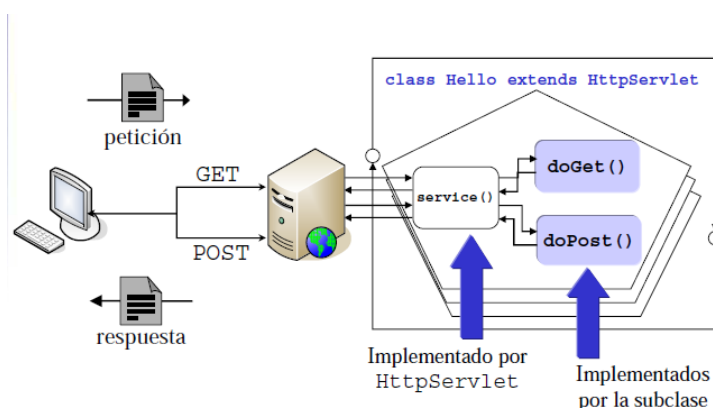
- El ciclo de vida de un servlet se inicia con la solicitud o insistencia, este es una copia del programa en versión ejecutable que se realiza en la primera petición, esto significa, si no existen instancias del servlet, el contenedor web carga la clase del servlet, crea una instancia e inicializa la instancia del servlet llamando a la función `init`.
- Al manejar sucesivas peticiones el contenedor crea un hilo que llama al método **service** de la instancia, este método **service** determina lo que ha llegado en la petición y llama a un método apropiado, puede ser `httpService`, este método nos indica la transferencia de archivos en formato HTML.
- La destrucción del servlet sucede cuando el contenedor toma la decisión de destruir totalmente al servlet llamando al método `destroy`⁷, este se utiliza cuando ya no se requiere más el *applet*⁸.

⁷ **Destroy.** Se llama cuando ya no se va a utilizar más el applet, cuando se necesita que sean liberados todos los recursos dispuestos por el applet.

⁸ **Applet.** Es un componente de una *aplicación* que se ejecuta en el contexto de otro programa, por ejemplo un navegador web

3.2.3 Consecuencias del Ciclo de Vida del Servlet

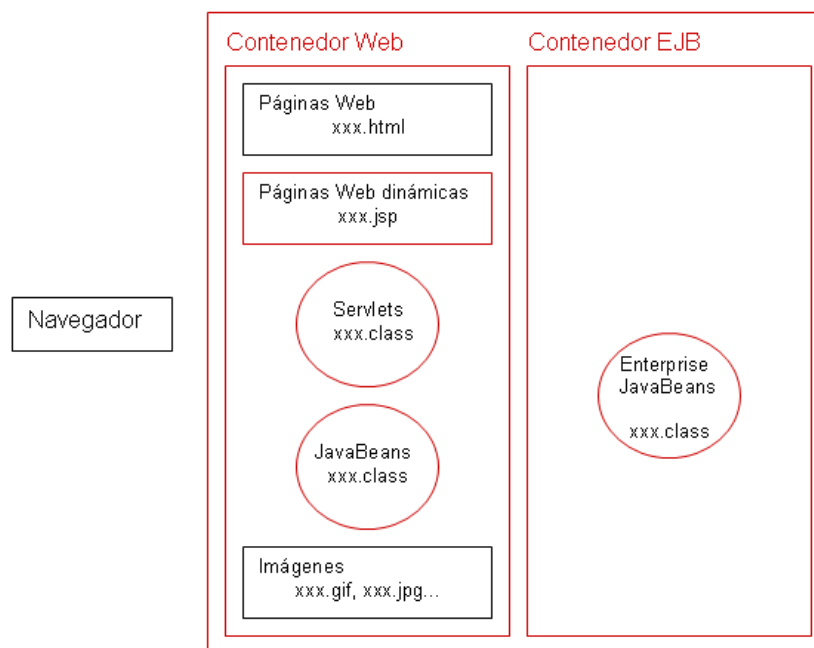
La maquina virtual, es donde se maneja el código nativo o ejecutable en una plataforma específica capaz de interpretar y ejecutar instrucciones expresadas en código binario especial generado por el compilador de java. Entre las ventajas de la maquina virtual tenemos: la compartición de datos entre varias instancias, la persistencia de las instancias que reducen el consumo de energía, eliminan los tiempos de inicialización e instanciación



Fuente: Web componet Development With Servlet and JSP Tecnologies Sum Microsystems

Figura 7: Funcionamiento de la Clase Extende HtttServlet

3.2.4 NetBeans 5.5



Fuente: Inetgiant.com

Figura 8: Arquitectura Java

NetBeans es un entorno de desarrollo, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio del año 2000 y continúa siendo el patrocinador principal de los proyectos.

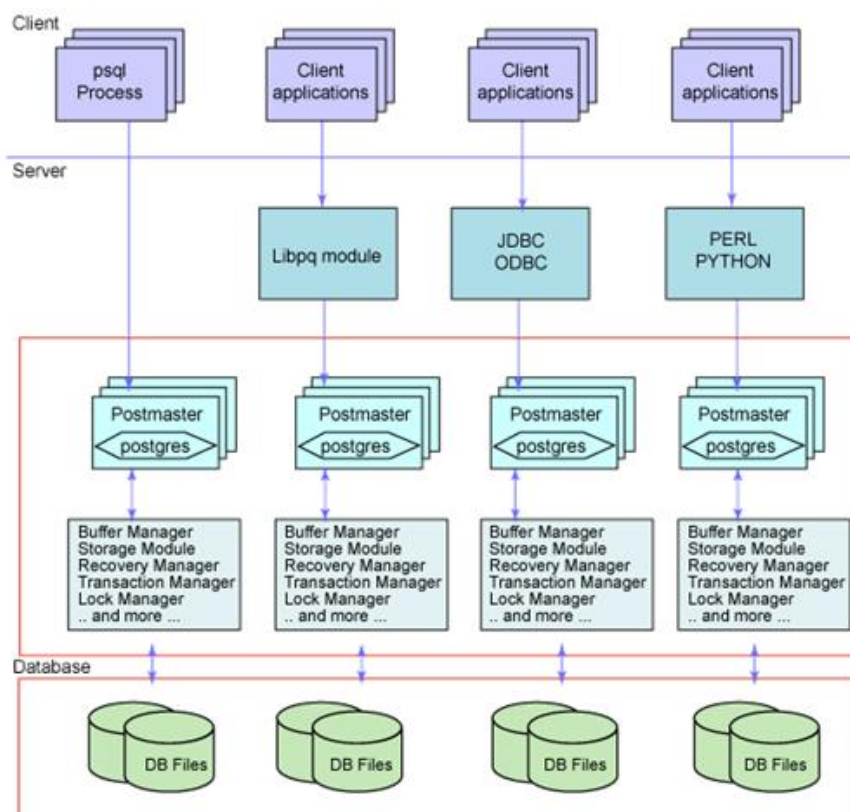
La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones

construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

3.3 Motor de Base de Datos PostgreSQL v 8.4

“PostgreSQL es un gestor de bases de datos orientadas a objetos (SGBDOO) muy conocido y usado en entornos de software libre, porque cumple los estándares SQL92 y SQL99, y también, por el conjunto de funcionalidades avanzadas que soporta, lo que lo sitúa al mismo o a un mejor nivel que muchos SGBD comerciales.” Richard Blum, 2007

PostgreSQL se distribuye bajo licencia BSD (distribución de software Berkeley), lo que permite su uso, redistribución, modificación con la única restricción de mantener el copyright del software a sus autores, en concreto, el PostgreSQL Global Development Group y la Universidad de California. PostgreSQL puede funcionar en múltiples plataformas (en general, en todas las modernas basadas en Unix, también en Windows de forma nativa).



Fuente: IBM.com

Figura 9: Arquitectura de PostgreSQL y los Procesos de Información General

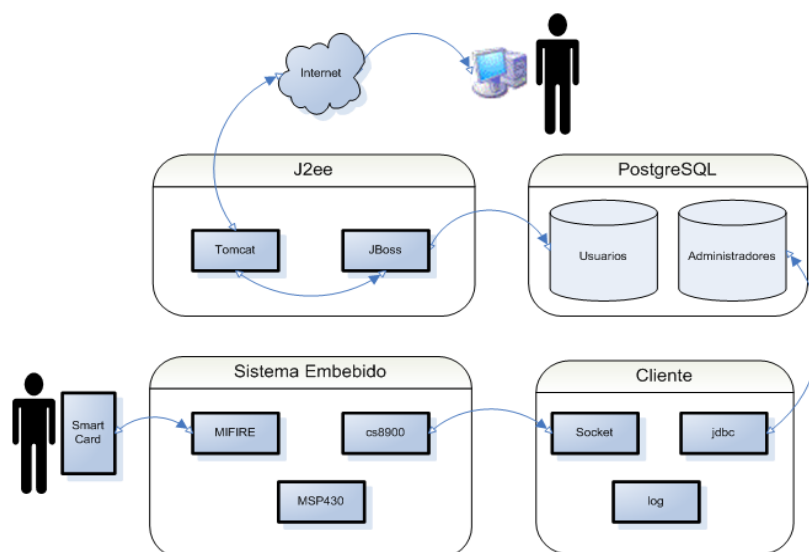
3.3.1 Breve Historia

La historia de PostgreSQL se inicia en el año 1986 con un proyecto del profesor Michael Stonebraker y un equipo de desarrolladores de la Universidad Berkeley (California), cuyo nombre original era POSTGRES. En su diseño se incluyeron algunos conceptos avanzados en bases de datos y soporte parcial a la orientación a objetos.

POSTGRES fue comercializado por Illustra, una empresa que posteriormente formó parte de Informix (que comercializaba el conocido *SGBD*⁹ del mismo nombre, recientemente absorbida por IBM y su DB/2). Llegó un momento en que mantener el proyecto absorbía demasiado tiempo a los investigadores y académicos, por lo que en el año 1993 se liberó la versión 4.5 y oficialmente se dio por terminado el proyecto.

En el año 1994, Andrew Yu y Jolly Chen incluyeron SQL en Postgres para posteriormente liberar su código en la web con el nombre de Postgres95. El proyecto incluía múltiples cambios al código original que mejoraban su rendimiento y legibilidad.

3.3.2 Prestaciones



Fuente: Profesores.elo.

Figura 10: Interacción con los Componentes de un Sistema

PostgreSQL destaca por su amplísima lista de prestaciones que lo hacen capaz de competir con cualquier SGBD comercial, así tenemos:

- Está desarrollado en lenguaje C, con herramientas como Yacc y Lex.

⁹ SGBD. Sistema de Gestión de Base de Datos

- La *API*¹⁰ de acceso al SGBD se encuentra disponible en lenguaje de programación C, lenguaje de programación C++, lenguaje de programación Java, lenguaje de programación Perl, lenguaje de programación *PHP*¹¹, lenguaje de programación Python y TCL, entre otros.
- Cuenta con un extenso conjunto de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario.
- Su administración se basa en usuarios y privilegios, dentro de los privilegios tenemos las restricciones propias del sistema lo que puede ser un administrador y un usuario común, con sus respectivos privilegios.
- Sus opciones de conectividad abarcan *TCP/IP*¹², sockets Unix y sockets NT, además de soportar completamente *ODBC*¹³. El término socket es también usado como el nombre de una interfaz de programación de aplicaciones (API) para el stock de protocolos de Internet TCP/IP, provista usualmente por el sistema operativo.

La base de datos PostgreSQL muestra altamente su confiabilidad en cuanto a estabilidad se refiere, puede extenderse con librerías externas para soportar encriptación, búsquedas por similitud fonética.

- Control de concurrencia multi-versión, lo que mejora sensiblemente las operaciones de bloqueo y transacciones en sistemas multi-usuario. A diferencia de la mayoría de otros sistemas de bases de datos que usan bloqueos para el control de concurrencia, Postgres mantiene la consistencia de los datos un

¹⁰ **API.** Una interfaz de programación de programa y aplicaciones. Conjunto de normas que determinan como debe usarse una determinada función de un programa en una aplicación.

¹¹ **PHP.** Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

¹² **TCP/IP.** Protocolo de control de transmisiones / Protocolo Internet. Es el protocolo estándar de comunicaciones en red utilizado para conectar sistemas informáticos a través de Internet

¹³ **ODBC.** (Open Database Connectivity) es un programa de interface de aplicaciones (API) para acceder a datos en sistemas manejadores de bases de datos tanto relacionales como no relacional

modelo multi-versión. Esto significa que mientras se consulta una base de datos, cada transacción ve una imagen de los datos (una versión de la base de datos) como si fuera tiempo atrás, sin tener en cuenta el estado actual de los datos que hay por debajo. Esto evita que la transacción vea datos inconsistentes que pueden ser causados por la actualización de otra transacción concurrente en la misma fila de datos, proporcionando aislamiento transaccional para cada sesión de la base de datos.

La principal diferencia entre multiversión y el modelo de bloqueo, es que en los bloqueos MVCC derivados de una consulta (lectura) de datos, no entran en conflicto con los bloqueos derivados de la escritura de datos y de este modo la lectura nunca bloquea la escritura y la escritura nunca bloquea la lectura.

- Soporte para vistas, claves foráneas, integridad referencial, disparadores, procedimientos almacenados, subconsultas y casi todos los tipos y operadores soportados en SQL92 y SQL99.

| Año | Nombre | Alias | Comentarios |
|------------|---------------|--------------|------------------------------------------------------------------------|
| 1986 | SQL-86 | SQL-87 | Primera publicación hecha por ANSI. Confirmada por ISO en el año 1987. |
| 1989 | SQL-89 | | Revisión menor. |
| 1992 | SQL-92 | SQL2 | Revisión mayor. |

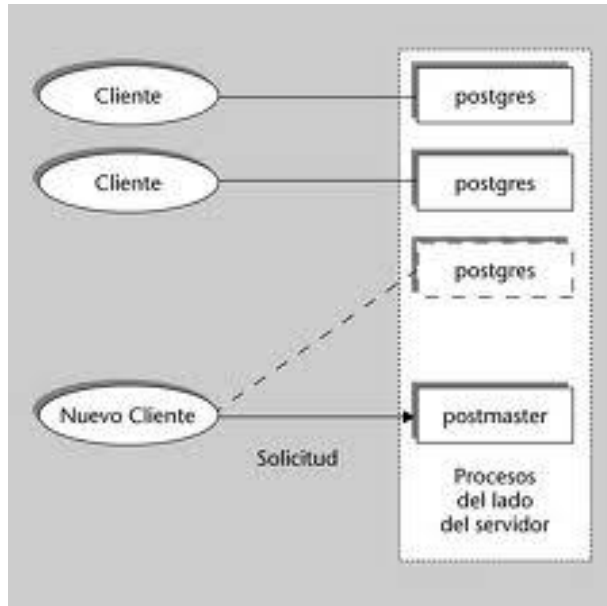
| | | | |
|------|----------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1999 | SQL:1999 | SQL2000 | Se agregaron expresiones regulares, consultas recursivas (para relaciones jerárquicas), triggers y algunas características orientadas a objetos. |
| 2003 | SQL:2003 | | Introduce algunas características de XML, cambios en las funciones, estandarización del objeto sequence y de las columnas auto numéricas. |
| 2006 | SQL:2006 | | ISO/IEC 9075-14:2006 Define las maneras en las cuales el SQL se puede utilizar conjuntamente con XML. Define maneras de importar y guardar datos XML en una base de datos SQL, manipulándolos dentro de la base de datos y publicando el XML y los datos SQL convencionales en forma XML. Además, proporciona facilidades que permiten a las aplicaciones integrar dentro de su código SQL el uso de XQuery, lenguaje de consulta XML publicado por el W3C (World Wide Web Consortium) para acceso concurrente a datos ordinarios SQL y documentos XML. |

| | | | |
|------|----------|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2008 | SQL:2008 | | Permite el uso de la cláusula ORDER BY fuera de las definiciones de los cursores. Incluye los disparadores del tipo INSTEAD OF. Añade la sentencia TRUNCATE. |
|------|----------|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------|

Fuente: IBM.com

Tabla 1.1: Operadores Soportados SQL 92 Y SQL 99

- Implementación de algunas extensiones de orientación a objetos. En PostgreSQL es posible definir un nuevo tipo de tabla a partir de otra previamente definida.



Fuente: Dataprix.com

Figura 11: Arquitectura de PostgreSQL

3.3.3 Limitaciones

Las limitaciones de este tipo de gestores de bases de datos suelen identificarse muy fácilmente. Entre las principales encontramos las siguientes:

- Puntos de recuperación dentro de transacciones. Actualmente, las transacciones abortan completamente si se encuentra un fallo durante su ejecución. La definición de puntos de recuperación permitirá recuperar mejor transacciones complejas.
- No soporta tablespaces para definir dónde almacenar la base de datos, el esquema, los índices. Un tablespace es una unidad lógica de almacenamiento dentro de una base de datos. Es un puente entre el sistema de ficheros del sistema operativo y la base de datos.
- El soporte a orientación a objetos, la herencia es el mecanismo fundamental para implementar la reutilización y extensibilidad del software. A través de ella los diseñadores pueden construir nuevas clases partiendo de una jerarquía de clases ya existente, que han sido comprobadas y verificadas, evitando con ello el rediseño, la modificación y verificación de la parte ya implementada. La herencia facilita la creación de objetos a partir de otros ya existentes, obteniendo características como pueden ser métodos y atributos, similares a los ya existentes.

4. BENEFICIOS DEL SISTEMA

El sistema se encarga del registro, control y reportes de las actividades de análisis químico, ingresos de calibraciones, sistema de agua potable, parámetros físicos, químicos, estándares para curvas de calibración, los reportes necesarios y las graficas en tiempo real. Estos registros permiten obtener la información mucho más rápido y sin redundancia en los datos.

Entre otros beneficios tenemos:

Una interfaz sencilla para el uso del personal de la empresa de agua potable EMAPA-I, especialmente para las personas que tienen pocos conocimientos del uso del computador.

Llevar un almacenamiento de los datos de manera más segura y siempre con respaldos, estos respaldos se los van a ubicar en el servidor principal de la empresa y siempre va a estar a disposición de los directivos y los administradores del sistema.

A través de los reportes que generan el sistema y el acceso en tiempo real a las graficas generadas en función de la información ingresada, es posible agilizar la toma de decisiones en el Laboratorio de Caranqui.

El sistema propuesto se encuentra realizado en aplicaciones actuales como son: el lenguaje de programación en *NetBeans*¹⁴ y una base de datos confiable como PostgreSQL que brinda gran seguridad en el almacenamiento de los datos, esto nos garantiza procesos rápidos y eficientes.

En tiempo actual todavía se realiza todo este proceso en papel, al implementar el sistema se lograra un ahorro considerable especialmente de papel y material de oficina ya que solamente se imprimirá los reportes cuando se considere necesario.

Con la disponibilidad del sistema en cualquier tiempo y en cualquier lugar, mediante el acceso al Internet, el personal de la empresa de agua potable EMAPA-I podrá realizar el control necesario del proceso del Laboratorio, y a su vez, incrementar el potencial de cada persona empleada en la empresa.

¹⁴ **NETBEANS.** Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java

Laboratorio de
Caranqui

I
FASE DE INICIO



1. VISIÓN

1.1 Propósito

El propósito de este documento es definir a alto nivel los requisitos de la aplicación Sistema de evaluación, control y reportes para el Laboratorio de Caranqui de EMAPA-I, parroquia de Caranqui utilizando herramientas libres.

El sistema se encargará de administrar el control de documentaciones, relacionadas con el proceso de generación de reportes, con el fin de tener una documentación organizada y de fácil acceso, que representa el medio que permite a las personas que colaboran en el Departamento de Sistemas conjuntamente con el Laboratorio de Caranqui, alcanzar los objetivos relacionados con el trabajo finalizado este proyecto con la entrega del sistema en el cuál va ser utilizado en beneficio de la institución.

1.2 Alcance

Este documento de visión se aplica al Sistema de evaluación, control y reportes para el Laboratorio de Caranqui de EMAPA-I, parroquia de Caranqui utilizando herramientas libres que es desarrollado por la Egresada Gabriela Patricia Jácome Quelal, de la Facultad de Ingeniería en Sistemas Computacionales de la Universidad Técnica del Norte.

Este sistema se encarga de la administración y el control del Laboratorio de Caranqui de manera integral; garantizando que el mismo sea general, justo, equitativo y progresivo; mediante una gestión coordinada y eficiente a fin de que el mismo sea de uso multiusuario, orientado al fortalecimiento del presupuesto municipal y así reinvertir los tributos en la ejecución de obras que viabilicen el desarrollo sustentable del cantón

La implementación del Sistema “Sistema de Evaluación, Control y Reportes para el Laboratorio de Caranqui de EMAPA-I, Parroquia de Caranqui Utilizando Herramientas libres estará basada en *JSP*¹⁵, contenida en el paquete de aplicaciones gratuitas.

El acceso al Sistema será a través de Internet Explorer o cualquier navegador desde un cliente, este último puede ser un sistema Windows XP Pro o Home, Windows 2000 Pro, Windows 2000 Server, Windows 2003, con sus respectivas actualizaciones, para permitir un acceso al Sistema se requiere de la configuración del Servidor Apache Tomcat.

Los Trabajadores de la empresa tendrán conocimientos del Windows básico, lo que les permitirá elaborar sus informes, evaluaciones, para lo cual se incluirá una función integrada en el sistema, la cual realiza todas estas tareas de una manera fácil y amigable para los usuarios.

1.3 Posicionamiento

- Mejoramiento de la Empresa

Como parte del mejoramiento de actualización tecnológica en la empresa EMAPA-I, gestionado por el departamento de informática en conjunto con el Laboratorio de Caranqui, se determina la creación del Sistema de evaluación, control y reportes para el Laboratorio de Caranqui de EMAPA-I, parroquia de Caranqui utilizando herramientas libres que permita mejorar las actividades relacionadas a la evaluación y poder llevar un control centralizado de los datos recolectados y almacenados en la Base de Datos generados en el Laboratorio de Caranqui.

¹⁵ **JSP**. Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo

- **Definición del Problema**

| | |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| El problema de | No cuenta con un sistema de registro y almacenamiento de forma eficiente y confiable de datos, debido a que en la actualidad todas las tareas se realizan en forma manual. |
| afecta a | Los diferentes laboratorios que pertenecen a la institución de EMAPA-I, involucrados en los procesos de almacenamiento y control de datos. |
| el impacto de este problema es: | Recopilar, almacenar y organizar información para el desarrollo de reportes y graficas de datos, para que estos sean accesibles de manera oportuna y eficaz desde lugares físicamente remotos a las instituciones de EMAPA-I. Este es un proceso automatizado con infraestructura para Web. |
| una solución exitosa debería | Automatizar el proceso, usando la intranet de la EMAPA-I, la misma que permite tener acceso al sistema aplicando interfaces amigables y sencillas desde cualquier punto que se encuentra ubicada la empresa, lo que permite manejar la |

| | |
|--|---------------------------------------------------|
| | información de una manera más confiable y rápida. |
|--|---------------------------------------------------|

Fuente: Propia

Tabla 1.2: Definición del Problema

- Declaración del Posicionamiento del Proyecto

| | |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Para | Autoridades de EMAPA-I. Personal del Laboratorio de Caranqui. Departamentos Sistemas. |
| Quien(es) | Funcionarios de EMAPA-I. |
| El (nombre del producto) | Sistema de evaluación control y reportes para el laboratorio de EMAPA-I, parroquia de Caranqui, utilizando herramientas libres. |
| Que hace. | Registra, almacena y despliega reportes de los datos generados por el Laboratorio de Caranqui. |
| Debido a que | El proceso actual no tiene ningún tipo de automatización todo el trabajo se lo realiza de manera manual. |
| Nuestro producto | Permite automatizar los diferentes procesos que implica el correcto control y resguardo de los datos del laboratorio EMAPA-I mediante una interfaz gráfica. Además proporciona un acceso inmediato y actualizado de la información desde cualquier punto que tenga acceso al |

| | |
|--|-----------------------------------------------|
| | sistema el cual nos permite generar reportes. |
|--|-----------------------------------------------|

Fuente: Propia

Tabla 1.3: Declaración del posicionamiento del proyecto

1.4 Descripción de los Interesados y Usuarios

Los interesados son todas aquellas personas directamente involucradas en la definición y alcance de este proyecto. A continuación se presenta la lista de los interesados:

- Descripción de los Interesados en este Proyecto

| NOMBRE | DESCRIPCION | RESPONSABILIDADES |
|---------------------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Dr. Carlos Játiva | Jefe de Laboratorio | Jefe del Laboratorio, el cual realiza actividades de control del Laboratorio de Caranqui. |
| Ing. Yolanda Prado | Jefe de Unidad de Recursos Informáticos | Administra el departamento de Recursos Informáticos. |
| Lic. Carlos Hidrobo | Jefe del Departamento de Sistemas | Responsable de coordinar con los diferentes usuarios la correcta determinación de los requerimientos y la correcta concepción del sistema. |
| Pablo Varela | Gerente de Mejoramiento Continuo | Responsable de coordinar con los jefes para el buen |

| | | |
|-----------------------------------|----------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| | | funcionamiento del sistema. |
| Egresada Gabriela Patricia Jácome | Analista de Sistemas | Quien realizara labores de gestión de requisitos, diseño de datos para posteriores versiones. Y las pruebas funcionales del sistema. |
| Usuarios | Manejo del Sistema. | Encargados del manejo del sistema y el ingreso de datos, personal del Laboratorio de Caranqui. |

Fuente: Propia

Tabla 1.4: Descripción de los Interesados del Proyecto

- Resumen de los Usuarios

Los usuarios son todas aquellas personas involucradas directamente en el uso del sistema. A continuación se presenta una lista de los usuarios:

- Entorno del Usuario

El personal del Laboratorio de Caranqui serán usuarios del sistema y por ende beneficiará así a EMAPA-I, ya que permitirá registrar y llevar un control de toda la información, generando reportes de manera inmediata.

Los usuarios entrarán al sistema identificándose sobre un ordenador con un sistema operativo Windows XP y tras este paso entrarán a la parte de aplicación diseñada para cada uno según su rol. Este sistema es similar a cualquier aplicación Windows y por

tanto los usuarios estarán familiarizados con su entorno.

Los informes serán generados con Microsoft Word, en formato PDF y de manera grafica.

El proceso de Registro, Control y Reportes está compuesto de las siguientes actividades:

LABORATORIO:

- Pantalla para análisis químico.
- Pantalla para dureza total.
- Pantalla para análisis físico.
- Pantalla para análisis microbiológico.
- Pantalla de cloro residual.
- Pantalla ingreso calibraciones.
- Pantalla de aviso y calibración de la curva.
- Pantalla para sistemas de agua potable.
- Catalogo de parámetros físicos.
- Catalogo de parámetros químicos,
- Catalogo de cloro residual.
- Catalogo de estándares para curvas de calibración.
- Reporte de curvas de calibración Nitratos, Nitritos, Amoniacos, Fosfatos.
- Reporte de resultados químicos.
- Reporte de resultados físicos.
- Reporte de resultados microbiológicos.
- Reporte de resultados calibración.
- Pantalla de Administración.

- Creación de perfiles
- Gráficos de cloro residual por sector de catalogo de cloro residual.
- Gráficos de cloro residual por todos los sectores del catalogo de cloro residual.
- Grafico PH por sistema.
- Grafico de turbiedad por sistema.
- Grafico de color por sistema.
- Grafico de coliformes totales por sistema
- Grafico E. colí por sistema.

1.5 Perfiles de los *Stakeholders*¹⁶

- **Coordinador del Proyecto**

| | |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Representante | Ing. Diego Ortiz |
| Descripción | Trabaja en la Universidad Técnica del Norte. Facultad Ciencias Aplicadas Especialidad Ingeniería en Sistemas Computacionales. |
| Tipo | Docente |
| Responsabilidades | Establecer los vínculos relacionales entre las autoridades de la Facultad con la Egresada Gabriela Jácome y la adecuada asesoría para poder obtener la culminación de su carrera. |
| Criterio de Éxito | Mantener una comunicación integral Tutor-Egresado, para realizar un |

¹⁶ **Stakeholder.** Cualquier persona interesada en, afectada por y/o implicada con el funcionamiento del sistema o software

| | |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | mejor desempeño en la creación del sistema para los Laboratorios de EMAPA-I. |
| Implicación | Revisor del adecuado funcionamiento del sistema y de la adecuada documentación del proyecto de Evaluación, control y reportes del los Laboratorios de EMAPA-I. |
| Entregable | N/A |
| Comentario | Mantener una relación constante con el desarrollo del proyecto. Brindar apoyo a nivel gerencial cuando sea necesario. |

Fuente: Propia

Tabla 1.5: Coordinador del Proyecto

- **Responsable del Proyecto**

| | |
|-------------------|---------------------------------------------------------------------------------------------------------------|
| Representante | Dr. Carlos Játiva. |
| Descripción | Jefe de Laboratorio. |
| Tipo | Empleado de EMAPA-I |
| Responsabilidades | Control de los Laboratorios de EMAPA-I |
| Criterio de Éxito | Salvaguardar en buen estado los Laboratorios conjuntamente con los empleados y las autoridades de la empresa. |

| | |
|-------------|--------------------------------------------------------------------------------------------|
| Implicación | Inspector del buen manejo de los Laboratorios. |
| Entregables | N/A |
| Comentarios | Es la persona encargada de llevar por un buen camino el funcionamiento de los Laboratorios |

Fuente: Propia

Tabla 1.6: Responsable del Proyecto

- **Responsable Funcional**

| | |
|------------------------|---------------------------------------------------------------------------------|
| Representante | Lic. Carlos Hidrobo. |
| Descripción | Jefe del Departamento de Sistemas de la empresa de agua potable EMAPA-I. |
| Tipo | Empleado. |
| Responsabilidades | Manejo correcto de los sistemas. |
| Criterio de Éxito | Correcta coordinación con los diferentes usuarios en la toma de requerimientos. |
| Grado de Participación | Activa |
| Comentario | Es la persona que se encarga de la buena relación entre todos los interesados |

Fuente: Propia

Tabla 1.7: Responsable Funcional

- **Perfiles de Usuario (Varios)**

| | |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Representante | Ing. Yolanda Prado |
| Descripción | Jefe de Unidad de Recursos Informáticos |
| Tipo | Empleado de la Empresa EMAPA-I |
| Responsabilidades | La administración del Departamento de Recursos informáticos. |
| Criterio de Éxito | Es la persona que se encarga de permitir la ejecución de los sistemas que benefician al buen funcionamiento de la Empresa de agua potable EMAPA-I. |
| Grado de Participación | Activa. |
| Comentario | Es la persona que da la disponibilidad para la utilización de los sistemas. |

Fuente: Propia

Tabla 1.8: Perfiles de los Usuarios

- **Perfiles de Usuario (Varios)**

| | |
|-------------------|-------------------------------------------------------------------------|
| Representante | Sr. Pablo Varela |
| Descripción | Gerente de Mejoramiento Continuo de la empresa de agua potable EMAPA-I. |
| Tipo | Empleado |
| Responsabilidades | Coordinador de los jefes departamentales. |

| | |
|------------------------|---------------------------------------------------------------------------------------------------------------|
| Criterio de Éxito | Buena comunicación con todos los jefes de los diferentes departamentos y extensiones de la empresa. |
| Grado de Participación | Activa |
| Comentario | Es el que se encarga de coordinar a todos los departamentos para obtener un beneficio dirigido a institución. |

Fuente: Propia

Tabla 1.9: Perfiles de los Usuarios

- **Necesidades de los Interesados y Usuarios**

| Necesidades del Cliente | Prioridad | Inquietudes | Solución Actual | Solución propuesta |
|---------------------------------------------------------------------------------------------|------------------|------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|---------------------------------------------------------------------------|
| Un sistema que facilite el procesamiento de los datos generados en Laboratorio de Caranqui. | Alta | El sistema debe registrar y almacenar la información para facilitar el manejo y control del sistema. | NO EXISTE. | Desarrollar el Sistema de almacenamiento y control de parámetros. |
| Realizar este sistema en el menor tiempo posible con el fin de ponerlo | Alta | Registrar la información de los Laboratorios. | Actualmente se realizan los registros del Laboratorio en forma | Registrar toda la información de los Laboratorios utilizando el sistema a |

| | | | | |
|-------------------------------------------------------------------------------------------------------------------|------|-----------------------------------------------------------------------------------|---------------------------|----------------------------------------------------------------------------------|
| en ejecución para el correcto Control y Manejo de la información. | | | manual. | implementar. |
| Elaborar un sistema utilizando herramientas informáticas que faciliten el procesamiento de la información. | Alta | Se debe utilizar las herramientas existentes o software libre para su desarrollo. | NO EXISTE | Desarrollar el sistema utilizando la herramienta como JSP, POSTGRESS y INTRANET. |
| La interfaz del sistema debe ser amigable, cumpliendo con todos los requerimientos establecidos por los usuarios. | Alta | Cumplir con todos los requerimientos de los usuarios. | NO EXISTE | La investigación y la colaboración. |
| Obtener | Alta | Tener un control en | Lo realizan de una manera | Generar reportes en OpenOffice y |

| | | | | |
|-----------|--|----------------------------------------------|--------|-------------------|
| Reportes. | | porcentajes de las muestras de agua | manual | de manera grafica |
|-----------|--|----------------------------------------------|--------|-------------------|

Fuente: Propia

Tabla 1.10: Necesidades de los Interesados y de los Usuarios

- **Alternativas y Competencia**

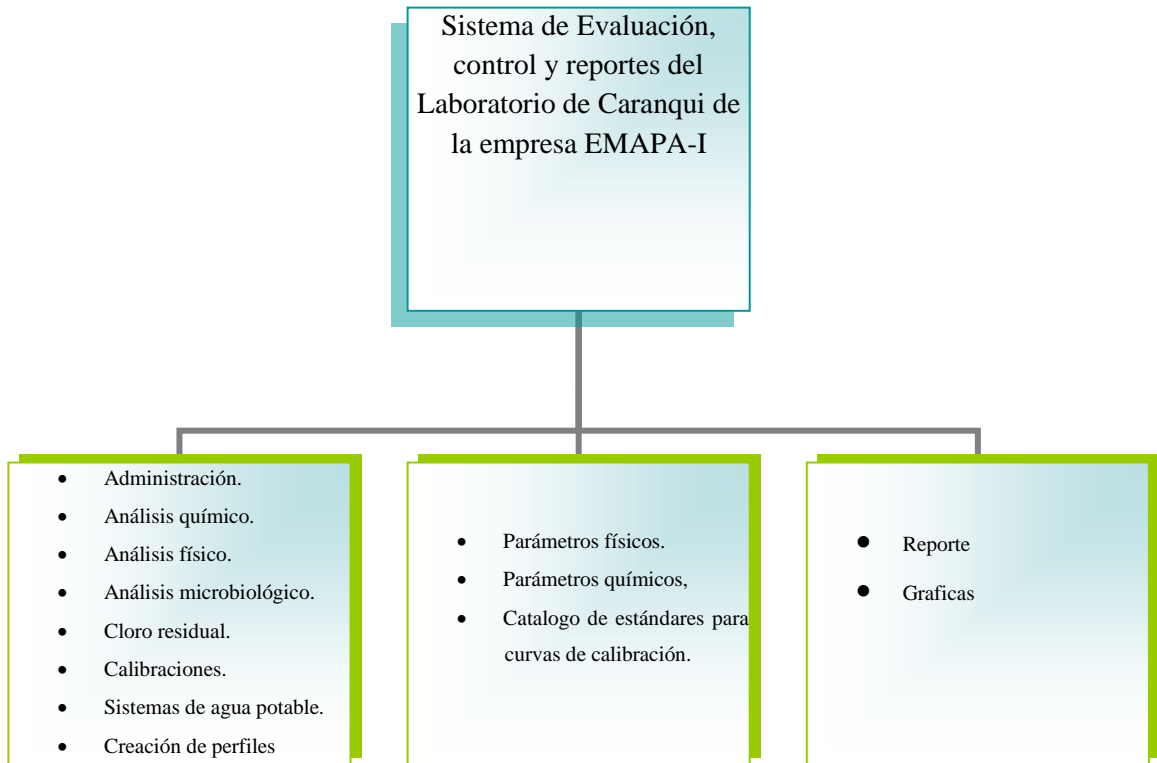
▪ **Adquirir un Sistema Desarrollado Externamente.**

En la busca de un sistema que este apto con todos los requerimientos proporciona por los usuarios, pero no ha sido factible por cuanto el esfuerzo de la implementación de estaciones de análisis automáticas. Involucran una alta inversión, ante esto la EMAPA-I ha optado por auspiciar la construcción del sistema que cumpla con todos los requerimientos proporcionados por los usuarios de la institución para garantizar un producto final de calidad.

1.6 Vista General del Producto

El producto a desarrollar es un sistema para la evaluación y Control de los datos del Laboratorio de EMAPA-I, con la intención de agilizar el desempeño de los empleados de la empresa y con información actual y real. Las áreas a tratar dentro del Laboratorio son los reportes que se generan al realizar la manipulación de los químicos en el agua y las calibraciones, en general la administración del Laboratorio y la administración del sistema esta se realizará a través de los directivos y personal autorizado por la empresa de agua potable EMAPA-I.

- **Perspectiva del Producto**



Fuente: Propia

Figura 12: Perspectiva del Software del Laboratorio

- **Resumen de Capacidades**

| Beneficios para el Usuario | Características |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Los tiempos de proceso se reducen | En el proceso que se realiza en el Laboratorio con la inclusión del sistema de evaluación que se implementara en la empresa EMAPA-I, va a realizarse de manera ordenada y sistematizada. |

| | |
|-------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Los usuarios van a poseer una interfaz amigable | El manejo del sistema se va realizar de la manera más fácil después de realizar la adecuada capacitación a los usuarios |
| Facilidad para el análisis | A través de los reportes generados por el sistema y las diferentes opciones que muestra el sistema. |
| Disponibilidad de reportes | Las personas que tengan acceso al sistema dispondrán de los reportes en el momento que ellos necesiten es decir estos reportes se generan en tiempo real. |

Fuente: Propia

Tabla 1.11: Resumen de Capacidades

- Suposiciones y Dependencias

Se asume que la Dirección de Sistemas de EMAPA-I donde se encuentre implantado el sistema, tenga acceso al servidor de base de datos y de aplicaciones a través de TCP/IP de la Empresa. Esto con el objetivo de que cada usuario pueda tener acceso al producto, y le permita consultar y generar reportes de acuerdo a sus privilegios dentro del sistema.

1.7 Costos y Precios

| DETALLE | | USD | REAL |
|-----------------|-------------------------------|------|------|
| Hardware | Equipos de Computación | 1500 | 1500 |
| | Servidor de Aplicación | | |
| | Web Servidor de Base de Datos | | |
| Software | NetBeans 5.5(OpenSours) | 0.00 | 0.00 |
| | Apache Tomcat | 0.00 | 0.00 |

| | | | |
|-----------------------------------------|----------------------------------------------------------|-------------|-------------|
| | 6.2(OpenSours) Base de Datos PostgreSQL(OpenSours) | 0.00 | 0.00 |
| Costo de Desarrollo | Pago por desarrollo | 1000 | 0.00 |
| Capacitación a los Desarrollares | Cursos y Libros, Asesoramiento | 1000 | 1000 |
| Proyecto | Papelería y Suministros de Oficina | 500 | 500 |
| Subtotal | (Parcial) | 4000 | 3000 |
| 5% Imprevistos | | 200 | 150 |
| Total | | 4200 | 3150 |

Fuente: Propia

Tabla 1.12: Costos y Precios

- **Instalación**

La instalación del producto es realizada por el personal de soporte del Departamento de Sistemas ya que es un sistema que utiliza tecnología Web.

1.8 Características del Producto

- **Facilidad de Acceso y Uso**

El Sistema de Evaluación, control y reportes del Laboratorio de Caranqui en la Empresa de EMAPA-I, es desarrollado utilizando Tecnología Web y las facilidades que ofrece la

herramienta *OpenSours*¹⁷ (Código Abierto), lo que permitirá fácil acceso y bajo costo para la adquisición de dichas herramienta y a la vez muestra una interfaz grafica y de fácil uso.

- **Mejor Control y Validación de la Información**

Los usuarios del Laboratorio de Caranqui contarán con facilidades para la verificación de la información y para tener control del manejo del sistema mediante los privilegios del sistema.

1.9 Restricciones

Este es un sistema diseñado especialmente para cubrir las necesidades del Laboratorio de Caranqui de la empresa de agua potable EMAPA-I, ya que en la actualidad todo este proceso lo generan sin el proceso de automatización es decir de manera manual.

- **Rangos de Calidad**

“Las siglas *RUP*¹⁸ en ingles significa Rational Unified Process (Proceso Unificado de Rational) es un producto del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo. Su meta es asegurar la producción del software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos.” Roger S Presuman. 2005

Dentro de los rangos de calidad el desarrollo del sistema se ajustará a la Metodología de Desarrollo de Software RUP, contemplando los parámetros de calidad que la metodología definida.

¹⁷**Open Source.** Es el software que está licenciado de tal manera que los usuarios pueden estudiar, modificar y mejorar su diseño mediante la disponibilidad de su código fuente

¹⁸**RUP.** Rational Unifiel Process.

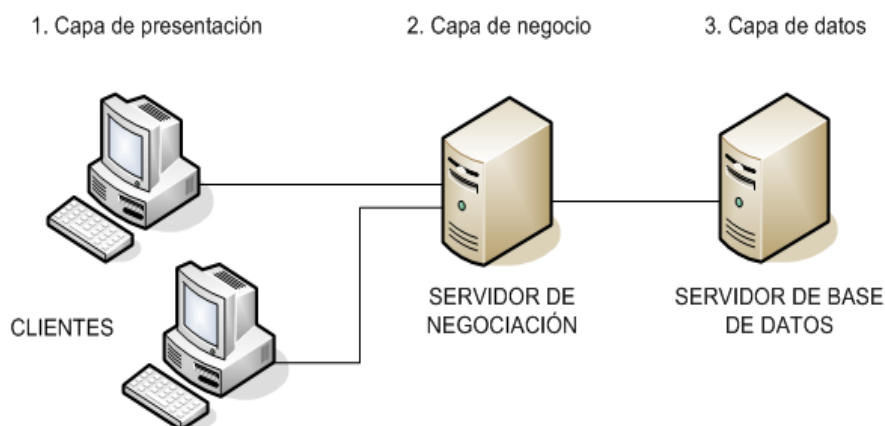
- Otros Requerimientos del Producto

Entre los requerimientos que se tiene que disponer es de un computador donde corran continuamente los servicios para la construcción de la aplicación.

-La intranet de la empresa de agua potable EMAPA-I, tiene las siguientes limitaciones:

1.10 Programación por Capas

“La **programación por capas** es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario. “ Ian Sommerville - 2005. Es decir la programación se divide en 3 partes fundamentales totalmente separadas y cualquier cambio que se realiza en alguna de ellas no interfiere al resto de capas.



Fuente: wikipedia.com

Figura 13.Programación por Capas

1.10.1 Introducción

Cuando se construye software como producto empresarial o comercial, se llevan a cabo varias técnicas de manera que el desarrollo se haga en forma ordenada y así poder asegurar un avance continuo del proyecto, un producto final de calidad, y además realizar posteriores mejoras sea una tarea más fácil.

Existen muchas prácticas de programación, dependiendo del tipo de software que se va a desarrollar y de la disciplina o disciplinas de programación que se utilicen en el desarrollo del producto.

Una de las más utilizadas se llama la programación por capas, que consiste en dividir el código fuente según su funcionalidad principal.

La programación para lograr sacarle el mayor provecho a la programación por capas se necesita seguir una serie de pasos complejos los cuales primeramente deben ser definidos para cada proyecto en específico, luego deben ser revisados para asegurarse de que el modelo adoptado cumpla con las normas necesarias para que la aplicación sea del agrado del usuario, y por último debe ser implementado por el grupo de desarrollo encargado para tal fin, los cuales siguiendo el modelo propuesto obtienen una herramienta útil para facilitar la labor de programación dividiendo la aplicación en módulos y capas fáciles de pulir.

1.10.2 Características de la Programación en Capas.

La programación por capas es una técnica de ingeniería de software propia de la programación por objetos, éstos se organizan principalmente en 3 capas: la capa de presentación o frontera, la capa de lógica de negocio o control, y la capa de datos.

Siguiendo el modelo, el desarrollador se asegura avanzar en la programación del proyecto de una forma ordenada, lo cual beneficia en cuanto a reducción de costos por tiempo, debido a que se podrá avanzar de manera más segura en el desarrollo, al ser

dividida la aplicación general en varios módulos y capas que pueden ser tratados de manera independiente y hasta en forma paralela.

Por otra parte, otra característica importante de recalcar es la facilidad para las actualizaciones de la aplicación. En este aspecto, la programación en capas juega un papel de suma importancia ya que sigue un estándar conocido en el ambiente de desarrollo de aplicaciones, lo cual da al programador una guía para hacer mejoras a la aplicación sin que esto sea una tarea tediosa y desgastante, siguiendo el estándar establecido para tal fin y dividiendo las tareas en partes específicas para cada capa del proyecto.

Las principales capas que siempre deben estar en este modelo son:

1.10.3 Capa de Presentación o Frontera

La presentación del programa ante el usuario, debe manejar interfaces que cumplan con el objetivo principal de este componente, el cual es facilitar al usuario la interacción con la aplicación. Para esto se utilizan patrones predefinidos para cada tipo de aplicación y para cada necesidad del usuario. La interfaz debe ser amigable y fácil de utilizar, ya que el usuario final es el que se va a encargar de utilizar el sistema y de dar retroalimentación al equipo de desarrollo en caso de que haya algo que mejorar.

Las interfaces deben ser consistentes con la información que se requiere, no se deben utilizar más campos de los necesarios, así como la información requerida tiene que ser especificada de manera clara y concisa, no debe haber más que lo necesario en cada formulario y por último, las interfaces deben satisfacer los requerimientos del usuario, por lo cual no se debe excluir información solicitada por el usuario final y no se debe incluir información no solicitada por el mismo.

Dentro de la parte técnica, la capa de presentación contiene los objetos encargados de comunicar al usuario con el sistema mediante el intercambio de información, capturando y desplegando los datos necesarios para realizar alguna tarea. En esta capa los datos se

procesan de manera superficial por ejemplo, para determinar la validez de su formato o para darles algún orden específico.

Esta capa se comunica únicamente con la capa de Reglas de Negocio o Control.

1.10.4 Capa de Lógica de Negocio o Control

Es llamada capa de reglas de negocio porque en esta se definen todas las reglas que se deben cumplir para una correcta ejecución del programa.

Es aquí donde se encuentra toda la lógica del programa, así como las estructuras de datos y objetos encargados para la manipulación de los datos existentes, así como el procesamiento de la información ingresada o solicitada por el usuario en la capa de presentación.

Representa el corazón de la aplicación ya que se comunica con todas las demás capas para poder llevar a cabo las tareas. Por ejemplo, mediante la capa de presentación obtiene la información ingresada por el usuario, y despliega los resultados. Si la aplicación se comunica con otros sistemas que actúan en conjunto, lo hace mediante esta capa. También se comunica con la capa de datos para obtener información existente o ingresar nuevos datos.

Recibe los datos que ingresó el usuario del sistema mediante la capa de presentación, luego los procesa y crea objetos según lo que se necesite hacer con estos datos; esta acción se denomina encapsulamiento.

Al encapsular los datos, el programa asegura mantener la consistencia de los mismos, así como obtener información precisa de las bases de datos e ingresar en las mismas, solamente la información necesaria, asegurando así no tener datos duplicados ni en las bases de datos, ni en los reportes solicitados por el usuario.

1.10.5 Capa de Datos

Es la encargada de realizar transacciones con bases de datos y con otros sistemas para obtener o ingresar información al sistema. El manejo de los datos debe realizarse de forma tal que haya consistencia en los mismos, de tal forma los datos que se ingresan así como los que se extraen de las bases de datos, deben ser consistentes y precisos.

Es en esta capa donde se definen las consultas a realizar en la base de datos, tanto las consultas simples como las consultas complejas para generación de reportes más específicos. Esta capa envía la información directamente a la capa de reglas de negocio para que sea procesada e ingresada en objetos según se necesite, esta acción se denomina *encapsulamiento*¹⁹.

1.10.6 Ventajas y Desventajas

La programación en capas no es una técnica rígida que debe implementarse solamente de una forma, sino que los desarrolladores de proyectos tienen múltiples maneras de implementarla según las tecnologías y tendencias que se utilicen.

La satisfacción de los requerimientos del usuario es la base para escoger el modelo de implementación a seguir. La tendencia a utilizar el modelo de programación en capas es grande cuando se trata principalmente de aplicaciones empresariales donde se deben manejar gran cantidad de subsistemas y módulos, así como generar reportes lo suficientemente complejos como para necesitar un orden estricto a la hora de desarrollar el proyecto.

Dentro del concepto de programación en capas, existen dos términos esenciales para el mejor entendimiento de los conceptos relativos a esta metodología, es aquí donde radica la importancia de la cohesión y el acoplamiento dentro de una aplicación generada mediante este método.

¹⁹ **Encapsulamiento.** se denomina **encapsulamiento** al ocultamiento del estado, es decir, de los datos miembro, de un objeto de manera que sólo se puede cambiar mediante las operaciones definidas para ese objeto

➤ **Cohesión:**

Este término es utilizado para describir el comportamiento que deben tener los módulos y objetos de un sistema o subsistema, comportamiento que describe la forma en que deben trabajar los objetos y módulos entre sí, con alta cohesión para que trabajando en conjunto los módulos y objetos puedan alcanzar un solo propósito de manera más eficaz y rápida.

Determina que las operaciones de un objeto deben trabajar en conjunto para alcanzar un propósito común. Es deseable que haya alta cohesión.

➤ **Acoplamiento:**

Se refiere al grado de dependencia que existe entre los módulos. Este grado de dependencia debe ser considerablemente bajo ya que el trabajo se divide en módulos para que cada uno tenga un funcionamiento específico y puede ser más factible la implementación por separado de cada uno. En caso de haber alto acoplamiento entre módulos no se estaría alcanzando el principal objetivo de este modelo, el cual es dividir una tarea grande en varias pequeñas, ya que los módulos actuarían como uno solo al estar altamente acoplados entre sí y se perdería el objetivo primordial de dividir el proyecto.

Laboratorio de
Caranqui

II
**PLAN DE
DESARROLLO DE
SOFTWARE**



2. PLAN DE DESARROLLO DE SOFTWARE

Este Plan de Desarrollo del Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto: Sistema de evaluación, control y reportes para el Laboratorio de Caranqui de EMAPA-I, utilizando herramientas libres.

Este documento provee una visión global del enfoque de desarrollo propuesto. El proyecto ha sido basado en una metodología de Unificación de Procesos con el fin de implantar un esquema inicial de ésta metodología para futuros desarrollos.

El enfoque desarrollo propuesto constituye una configuración del proceso Unificación de Procesos de acuerdo a las características del proyecto, seleccionando las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los *artefactos*²⁰ de la Unificación de Procesos.

2.1 Propósito

El propósito del Plan de Desarrollo de Software es proporcionar la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo del Software.

Los usuarios del Plan de Desarrollo de Software son:

El jefe del proyecto lo utiliza para organizar, controlar y generar reportes de todas las tareas realizadas en el Laboratorio de agua potable y en general realizar su seguimiento.

El desarrollador lo usa para entender lo qué debe hacer, cuándo debe hacerlo y qué otras actividades dependen de ello.

²⁰ **Artefactos.** Elementos materiales que los humanos han construido o modificado

2.2 Alcance

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo del Sistema de Evaluación, control y reportes del Laboratorio de EMAPA-I. Durante el proceso de desarrollo en el artefacto “Visión” se definen las características del producto a desarrollar. Para el Plan de Desarrollo del Software, me he basado en la captura de requerimientos por medio del stakeholder del Laboratorio para hacer una estimación aproximada, una vez comenzado el proyecto y durante la fase de Inicio se generará la primera versión del sistema “Visión”, el cual se utilizará para refinar este documento. Posteriormente, el avance del proyecto y el seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento produciendo nuevas versiones actualizadas.

2.3 Vista General del Proyecto

- ✓ La información que a continuación se incluye ha sido extraída de las diferentes reuniones que se han celebrado con los diferentes stakeholders desde el inicio del proyecto.
- ✓ Mediante la inclusión de sistemas informáticos en el Laboratorio de agua potable EMAPA-I permite mejorar los ingresos con parámetros reales y generar los reportes oportunos en tiempo real.
- ✓ Concluir con los requerimientos pertinentes para el buen funcionamiento del Laboratorio.
- ✓ El proyecto debe proporcionar una adecuada administración y control interno del Laboratorio.

Suposiciones y Restricciones

Las suposiciones y restricciones respecto del sistema, y que se derivan de las entrevistas con los stakeholder son:

- ✓ El proyecto debe ser, en lo posible, adaptable para el uso de los Laboratorios de la empresa EMAPA-I tomando como referencias todas las sugerencias vertidas por los stakeholder.
- ✓ Se debe examinar que el sistema sea seguro, brindando defensa a la información y confianza en la difusión de los datos.
- ✓ Debe tenerse en cuenta las limitaciones de la tecnología y de los conocimientos de los usuarios respecto al manejo del sistema. En la mayoría de los casos existe un computador de capacidad media y poco conocimiento del manejo del computador, por lo que en la etapa de capacitación se pondrá realce a la situación del manejo de la tecnología por parte de los usuarios.

Entregables del proyecto

Es preciso destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de permanencia y estabilidad de los artefactos.

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos propuesta para este proyecto.

- **Plan de Desarrollo de Software**

Es el presente documento.

- **Visión**

Esta documentación precisa la visión general del proyecto desde el aspecto de los usuarios, detallando los requerimientos establecidos por las personas involucradas.

- **Glosario**

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología utilizada.

- **Especificaciones de Casos de Uso**

Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: pre-condiciones, pos condiciones, flujo de eventos, requisitos no-funcionales asociados. También, para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Actividad.

Prototipos de Interfaces de Usuario

Se trata de prototipos que permiten a los usuarios tener una breve idea de lo que va hacer el manejo del sistema cuando ya se encuentre implementado, en tal caso es para poder realizar las falencias o los vacios que tiene el sistema y tratar de cubrir todos los requerimientos establecidos en la primera etapa y los requerimientos establecidos una vez realizado las pruebas del primer prototipo.

Estos prototipos van desde la primera etapa del proyecto, desde la utilización de papel para realizar de manera grafica los procesos del sistema, seguidamente utilizando las herramientas tecnológicas o herramientas graficas, siguiendo con el avance del proyecto hasta llegar a la culminación con todos los procesos realizados después de haber obtenido esta información de parte de los usuarios.

- **Modelo de Análisis y Diseño**

Dentro del modelo de análisis y diseño se encuentra el estudio o la preparación adecuada para poder llegar al estado de culminación del proyecto, dentro de lo que es el diseño se busca una manera fácil y agradable para el manejo que tiene que realizar el usuario en la manipulación del proyecto.

- **Modelo de Datos**

Dentro del modelado de datos se encuentra en el modelado de la base de datos esto describe la representación lógica de los datos, de acuerdo con el enfoque de lo que es modelo relacional. Para expresar este modelado se utiliza un diagrama de clases utilizando el modelado *UML*²¹.

- **Modelo de Implementación**

Este modelo se refiere a la implementación del sistema en sí, es decir la programación de componentes incluyendo todo lo que refiere a ficheros, código fuente y todo lo que se refiere al despliegue del sistema.

- **Modelo de Componentes**

Este modelo muestra los componentes que forman parte del sistema.

²¹ **UML.** Lenguaje Unificado de Modelado

- **Casos de Prueba**

Dentro de los casos de prueba se aplican las pruebas de regresión. Cada caso de prueba llevara asociado un procedimiento de prueba con las debidas instrucciones y siempre y cuando depende del tipo de prueba y el procedimiento siempre tiene que ser automatizable mediante script de pruebas.

- **Lista de Riesgos**

Siempre debe existir una lista de riesgo en los documentos vigentes en el proyecto, ordenadamente.

- **Manual de Instalación**

Este documento incluye las instrucciones para realizar la instalación del producto.

- **Material de Apoyo al Usuario Final**

Dentro del material de apoyo se encuentra toda la documentación para prestar la debida ayuda al usuario para tener un correcto manejo del sistema implementado en la empresa EMAPA-I

- **Producto**

Dentro de lo que es Producto ya realizado se lo tendrá que almacenar en dispositivos adecuados como pueden ser DVD, CD, con todos los instrumentos y material necesarios para poder realizar una adecuada y perfecta instalación del sistema y para su correcto funcionamiento.

2.4 Organización del Proyecto

▪ Participantes en el Proyecto

Dentro de los participantes del proyecto se incluirán a las siguientes personas que se encuentran involucradas en la realización del sistema sin olvidar de las personas que estuvieron presentes en la etapa de Inicio, de Elaboración y la fase de construcción.

La elaboración de estas etapas será finalizada con la colaboración de las siguientes personas y sus dependencias respectivas:

▪ Jefe de Proyecto.

Con una experiencia en el manejo de la empresa presta la ayuda pertinente para obtener todos los requerimientos y así dar fin a la etapa de inicio.

▪ Analista de Sistemas.

El perfil establecido es: Ingeniero en Informática con conocimientos del funcionamiento de los Laboratorios de EMAPA-I, y de los sistemas existentes en la empresa.

▪ Analistas – Programadores.

Que posea conocimientos en el desarrollo de proyectos informáticos, con el fin de obtener un buen resultado en la creación del prototipo para el Laboratorio de EMAPA-I, Esta tarea es encomendado a la Sta. Egresada Gabriela Patricia Jácome Quelal.

- **Ingeniero de Software.**

El perfil establecido es: Ingeniero en Informática que tendrá la tarea o labor de recoger todos los requerimientos, configuración de los equipos, documentación pertinente y el diseño de datos. A la vez será la encargada de las pruebas funcionales en la Empresa.

- **Interfaces externas**

Están involucrados los diferentes participantes para la realización del proyecto, esto va desde la obtención de requerimientos hasta la evaluación del buen funcionamiento del sistema y todas sus fases intermedias.

La analista de sistemas Egda. Gabriela Jácome, interactuará activamente con todas las personas involucradas, especialmente con el personal de la empresa de agua potable EMAPA-I, para poder dar cumplimiento a la creación y la validación del sistema.

- **Roles y Responsabilidades**

A continuación se describen las principales tareas de cada persona involucrada en las etapas de Inicio y Elaboración de acuerdo con los roles que presta para la culminación de este proyecto.

| PUESTO | RESPONSABILIDAD |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Jefe del Proyecto | La persona que se encuentra de jefe de proyecto es el encargado de designar las tareas correspondientes a todos los colaboradores para llegar exitosamente a la culminación del proyecto, a la vez mantendrá la comunicación con todo |

| | |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | el personal para así poder cumplir los objetivos establecidos al comienzo de este proyecto, además se encargara de planificar, supervisar y controlar el proyecto |
| Analista de Sistemas | Es la persona encargada de la obtención de requerimientos mediante reuniones con las personas que se encuentran directamente involucradas con el desarrollo del software, es decir con las personas que van a manipular el sistema. También es el encargado del diseño general del sistema |
| Programador | Es la persona encargada de la programación del prototipo y la evaluación de las pruebas para llegar a la validación directa con el usuario |
| Ingeniero de Software | Es la persona encargada de gestionar la obtención de requerimientos, gestionar todo lo que se refiere a la configuración, elaboración del modelado del sistema, programar las pruebas funcionales, documentar todo el proceso, y poner en despliegue el sistema. |

Fuente: Propia

Tabla 2.1: Perfiles de los Usuarios

2.5 Plan del Proyecto

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

- Plan de las Fases

El desarrollo de este proyecto tiene su inicio desde la obtención de la información, requerimientos necesarios, hasta llegar a la culminación del proyecto, con la documentación complementaria y la instalación del sistema en la empresa de agua potable EMAPA-I, teniendo en cuentas las etapas intermedias del proyecto como son la elaboración y la construcción. A continuación se mostrara el tiempo estimado en cada fase.

| FASE | DURACION (Semanas) |
|--------------|-------------------------------|
| Inicio | 1 Mes |
| Elaboración | 4 Meses |
| Construcción | 5 Meses |
| Transición | ~ |

Fuente: Propia

Tabla 2.2: Plan de Fases

Para tener una mejor visión del sistema a cerca de las fases tenemos:

| FASE | DESCRIPCION |
|----------------|---------------------------------------------|
| Fase de Inicio | Dentro de esta fase se establecen todos los |

| | |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>requerimientos establecidos desde el punto de vista de los usuarios, los cuales serán tomados en cuenta para la creación del sistema, ya que los usuarios son los concedores del funcionamiento del Laboratorio. Los principales casos de uso serán identificados en el Plan de Desarrollo del Proyecto. La aceptación del cliente / usuario y el Plan de Desarrollo marcan el final de esta fase.</p> |
| Fase de Elaboración | <p>Una vez culminada la fase de inicio empezamos con la fase de elaboración esto se refiere al análisis de los requerimientos y el desarrollo de un prototipo. Al final de esta fase todos los casos de uso oportunos a los requerimientos que serán realizados en la fase de Construcción deben estar analizados y diseñados. La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. Dentro de esta fase tendrá como objetivo la identificación y especificación de los principales casos de uso, así como su realización preliminar en el Modelo de Análisis /Diseño.</p> |
| Fase de Construcción | <p>Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis/Diseño. El producto se construye en base a 2 o 3 pruebas dentro de esta fase, cada una produciendo una mejora a la cual se le aplican las pruebas y se valida con el cliente/usuario. En esta fase comienza la elaboración de material de apoyo al usuario. La pauta que marca el fin de esta fase es la versión del</p> |

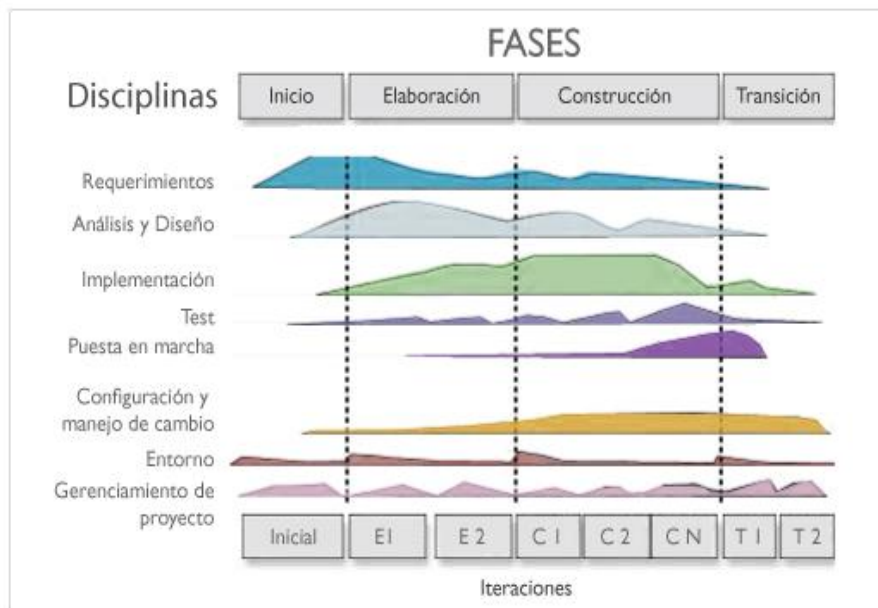
| | |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | producto con la capacidad operacional parcial para realizar pruebas. |
| Fase de Transacción | En esta fase se prepara la implantación y cambio del sistema previo de manera adecuada, incluyendo la capacitación a los usuarios del Laboratorio de Caranqui. Para marcar el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, |

Fuente: Propia

Tabla 2.3: Descripción de las Fases

- Calendario del Proyecto

A continuación se presenta un calendario de las principales tareas del proyecto incluyendo sólo las fases de Inicio y Elaboración. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto. La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina en un momento determinado del desarrollo.



Fuente: epidataconsulting.com

Figura 14: Fases de un Proyecto

- **Seguimiento y Control del Proyecto**

▪ **Gestión de Requisitos**

Cada requisito será evaluado por sus diferentes atributos: importancia, estado, iteración y donde se implementa cada requerimiento. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito. Los cambios en los requisitos serán tomados en cuenta para el mejoramiento del sistema y en sí beneficiara directamente a la empresa de agua potable EMAPA-I, siempre y cuando pasen la adecuada evaluación y análisis.

▪ **Control de Plazos**

El calendario del proyecto tendrá un seguimiento por el jefe de proyecto y por el director de tesis.

▪ **Control de Calidad**

Con respecto al control de calidad utilizamos la herramienta RUP para realizar un análisis y determinar el estado que se encuentra la elaboración del proyecto y su continua revisión que se la hace con la implementación de esta herramienta.

▪ **Gestión de Riesgos**

A partir de la fase de Inicio se mantendrá una lista de riesgos asociados al proyecto y de las acciones establecidas de contingencia.

| # | Descripción del Riesgo | Impacto | Probabilidad de Ocurrencia | Estrategia de mitigación del riesgo |
|---|-------------------------------------------------------------------------------------------------------------------------------|---------|----------------------------|-----------------------------------------------------------------------------------------------------------|
| 1 | El sistema de ingreso y control del Laboratorio de EMAPA-I podría no estar listo para el mes en que se planifica la entregar. | 7 | 30% | Monitorear constantemente el progreso y el cumplimiento de metas en el cronograma e incrementar esfuerzo. |
| 2 | Requerimientos nuevos para aumentara a la implementación del sistema una vez realizado el diseño. | 5 | 20% | Determinar si el sistema está apto para realizar cambios de requerimientos. |
| 3 | Falta de participación de los usuarios finales | 9 | 10% | Motivar a los usuarios finales del sistema |

| | | | | |
|---|--------------------------------------------------------|---|-----|---------------------------------------------------------------------------------------------------------------------------------------------|
| | | | | hacerles conocer que cualquier aporte que ellos den para el desarrollo del sistema es de gran importancia para la realización del proyecto. |
| 4 | Cambios de autoridades en la empres auspiciante | 7 | 10% | Realizar nuevos contactos y explicaciones con las autoridades pertinentes |
| 5 | Incompatibilidad con los sistemas que posee la empresa | 6 | 40% | Utilizar una sola plataforma en todas las maquinas de los clientes. |

Fuente. Propia

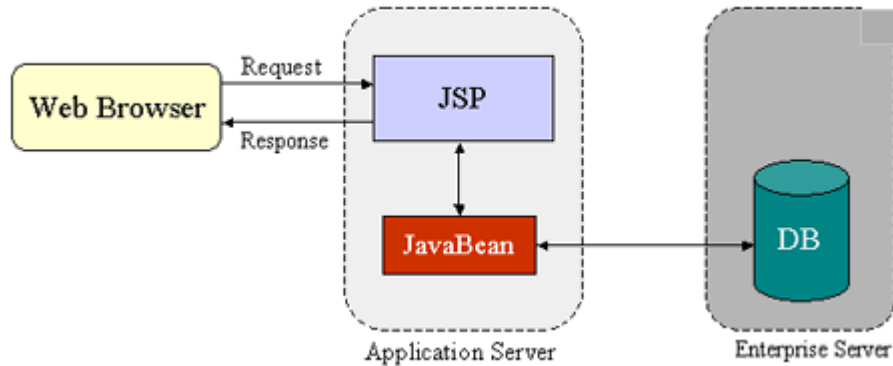
Tabla 2.4: Gestión de Riesgos

Laboratorio de
Caranqui

III
TECNOLOGIA
JSP



3. TECNOLOGIA JSP



Fuente: Sr. Ramos-Desarrollo Java

Figura 15: Tecnología JSP

3.1 Introducción

JSP son las siglas de Java Server Pages, que en castellano vendría a decir algo como páginas de servidor JAVA, que es una tecnología orientada a crear aplicaciones web con programación en java. “Java Server Pages es una tecnología basada en la plataforma Java 2 que simplifica el proceso de desarrollo de sitios web dinámico. Con JSP tanto los desarrolladores como los diseñadores Web utilizando código Java y una serie de etiquetas especiales determinadas.” Mc Graw – Hill, Manual de referencia JSP

En las JSP se escribe el texto que va a ser devuelto en la salida, normalmente código *HTML*²² incluyendo código java dentro del para poder modificar o generar contenido dinámicamente. El código java se incluye dentro de las marcas de etiqueta %y%

3.2 Antecedentes.

²² **HTML**. Las siglas de HyperText Markup Language («lenguaje de marcado de hipertexto»), es el lenguaje de marcado predominante para la elaboración de páginas web.

En el mundo cambiante en el cual vivimos los avances tecnológicos han revolucionado a la sociedad entera y es así que estos avances impulsan el progreso en diferentes áreas, dentro del campo de programación se produjo una mejora significativa de la API servlets de Java que fue seguida por el API de JSP esta tecnología lleva todo el potencial de Java al servidor web. Con conectividad a base de datos, acceso a trabajo en red, operaciones de subprocesos múltiples y sobre todo un modelo de proceso diferente

La oleada inicial de Java en clientes en forma de applets fue muy popular pero causó decepción en la práctica. La utilidad de los applets estaba limitada por el considerable número de incompatibilidades entre navegadores por los excesivos periodos de descarga con módems lentos y por restricciones de seguridad. Java es un servidor que no sufre de restricciones del entorno applets. No aparecen inconsistencias del navegador por que no es necesario que este posea una maquina virtual da Java.

El navegador solo tiene que generar HTML, las precipitaciones de seguridad se limitan a aquellas ya gestionadas por el servidor web que esta normalmente en un entorno cerrado con controles separados.

3.3 Características.

Las paginas JSP se ejecutan en un componente de servidor conocido como contenedor de JSP que las traduce a servlets Java equivalente. Por esta razón los servlets y las paginas JSP están íntimamente relacionadas. Las tecnologías servlets y JSP de Java, no se plantean como dos alternativas a poder utilizar separadamente, sino como técnicas complementarias. Es más las paginas JSP cuando se compilan se transforman en servlets.

Con JSP se puede crear aplicaciones web que se ejecuten en varios servidores web, de múltiples plataformas, ya que Java es la esencia un lenguaje multiplataforma.

Las paginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts ejecutables en el servidor de sintaxis Java. Por lo tanto las JSP pueden ser escritas con cualquier editor HTML/XML habitual.

La característica ofrecida por JSP como alternativas a la generación de contenido dinámico para la web se puede asumir en:

MEJORAS EN EL RENDIMIENTO.

- Utilización de procesos ligeros (hilos Java) para el manejo de las peticiones.
- Manejo de múltiples peticiones sobre una paginas JSP en un instante dado.
- El contenedor servlets puede ser ejecutado como parte del servidor web.
- Facilidad para compartir recursos entre peticiones (hilos con el mismo padre: servlet container).

SOPORTE DE COMPONENTES REUTILIZABLES

- Creación, utilización y modificación de JavaBeans del servidor.
- Los JavaBeans utilizados en paginas JSP puede ser utilizados en servlets, applets o aplicaciones Java.

SEPARACION ENTRE CODIGO DE PRESENTACION Y CODIGO DE IMPLEMENTACIÓN.

- Cambios realizados en el código HTML relativos a como son mostrados los datos, no interfiere en la lógica de programación y viceversa. Las paginas JSP tienen todas las ventajas de los servlets y además poseen ventajas propias.
- Se vuelven a compilar automáticamente cuando es necesario.
- Como están en espacio común de documentos de servidor web, dirigirse a ellas es más fácil que dirigirse a los servlets.

- Como las paginas JSP son similares a HTML tiene mayor compatibilidad con las herramientas de desarrollo web.

Los servlets y Java Server Pages son dos métodos de creación de páginas web dinámicas en servidor usando el lenguaje Java. En ese sentido son similares a otros métodos o lenguajes que generan páginas web en el servidor. Sin embargo se diferencia de ellos en otras cosas.

Para empezar, los JSP y servlets se ejecutan en una maquina virtual Java lo cual permite que en el principio se pueda usar en cualquier tipo de ordenador, siempre que exista una maquina virtual Java. Cada servlets o JSP se ejecuta en su propia hebra, es decir en su propio contexto, pero no se comienza a ejecutar cada vez que recibe una petición sino que persiste de una petición a la siguiente de forma que no se pierde tiempo en invocarlo. Su persistencia le permite también hacer una serie de cosas de forma más eficiente como por ejemplo: conexión a base de datos y manejo de sesiones.

El problema de utilizar servlets directamente es que aunque son muy eficientes son muy tediosas de programar puesto que hay que generar la salida del código HTML con gran cantidad de función println. Este problema se resuelve fácilmente utilizando JSP, puesto que aprovecha la eficiencia del código Java para generar el contenido dinámico y la lógica de presentación con HTML normal.

Cuando en una página JSP se necesita introducir mucha funcionalidad, es decir, implantar mucho código Java para generar el contenido dinámico de la página, ese mismo hecho lleva a que el código de las paginas JSP no sea demasiado claro.

El dilema esta en decidir cuándo utilizar servlets y cuando JSP. Lo ideal sería usar JSP cuando el dinamismo que se pretende no supone introducir mucho código Java en las páginas, puesto que esto oscurecería el código. Sin embargo, cuando hay mucha funcionalidad y necesitamos mucho código Java lo ideal sería utilizar una página JSP que llamarse a un servlets que contenga la funcionalidad necesaria para que este realice

el trabajo y genere la respuesta, ocupándose el código JSP de presentar la información que devuelve al servlets.

3.4 JSP

Java

“Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa”. Harvey M. Deitel, Paul J. Deitel, Guillermo Trujano Mendoza - 2004

En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Fue desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

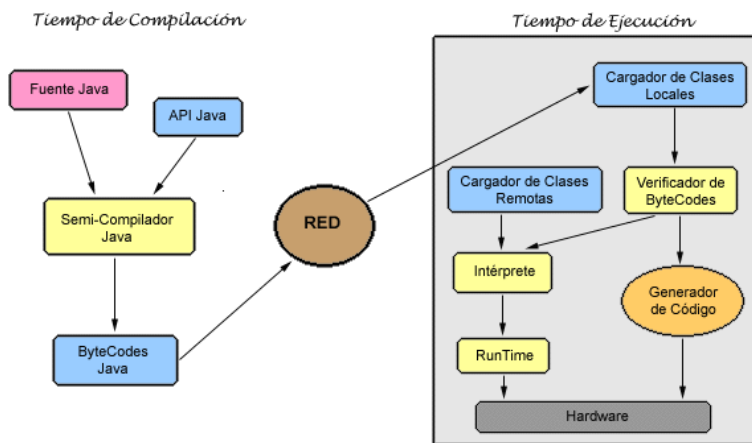


Fuente: ingmmurillo.blogspot.com

Figura16: Estructura de un Proyecto Java

Java compila el código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecuta.

Cualquier plataforma donde exista una JVM se puede ejecutar el código objeto independientemente de la máquina en el que ha sido generado.

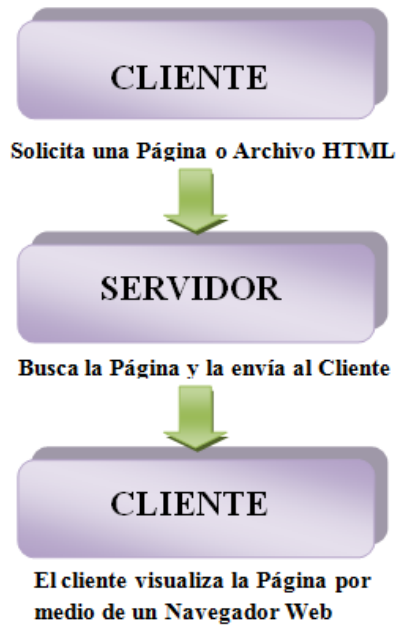


Fuente: mproject.com

Figura17: Compilación del Programa en Diferentes Tiempos.

HTML

“HTML es el lenguaje con el que se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web”. Javier Eguíluz Pérez 2004. Es decir por medio del lenguaje HTML, podemos navegar por miles y miles de páginas a través de la WWW. Es un lenguaje que sirve para escribir hipertexto, es decir, documentos de texto presentado de forma estructurada, con enlaces (links) que conducen a otros documentos o a otras fuentes de información que pueden estar en la propia máquina o en máquinas remotas de la red.

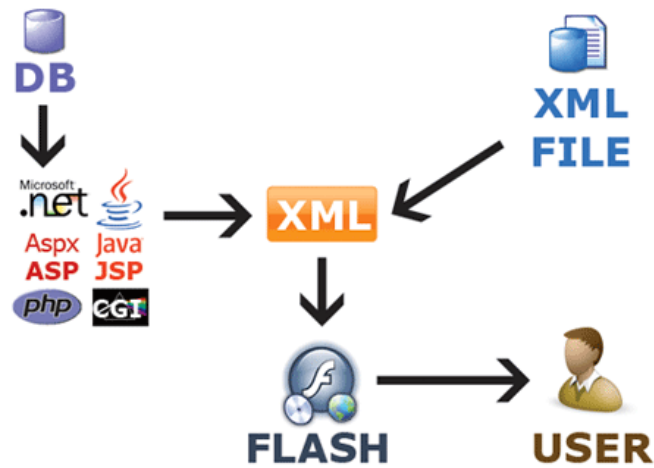


Fuente: Propia

Figura18: Funcionamiento de HTML.

XML

“XML es un lenguaje de metamarcado que ofrece un formato para la descripción de datos estructurados. Esto facilita unas declaraciones de contenido más precisas y unos resultados de búsquedas más significativos en varias plataformas.” Johnny Brochard – 2001. Además, XML habilitará una nueva generación de aplicaciones para ver y manipular datos basadas en el Web.



Fuente: Bulmaro Nogera-culturización.com

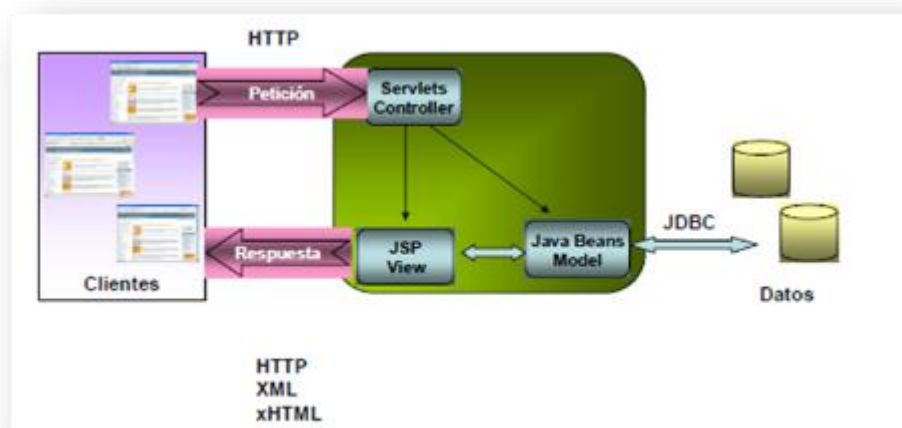
Figura19: Funcionamiento de XML.

Diferencia entre XML Y HTML

XML y HTML son lenguajes muy diferentes. Ambos nacen de la misma inspiración por lo que su sintaxis es similar, aunque cada uno fue diseñado para cumplir distintas funciones:

En forma resumida el XML sirve para describir información y el HTML sirve para darle formato y presentarla a través de un navegador. Es decir que el XML no es ni será nunca un reemplazo del HTML sino un complemento que sirve para manejar la

información separada del formato.



Fuente: export.writer.zoho.com

Figura 20: Modelo de Arquitectura JSP

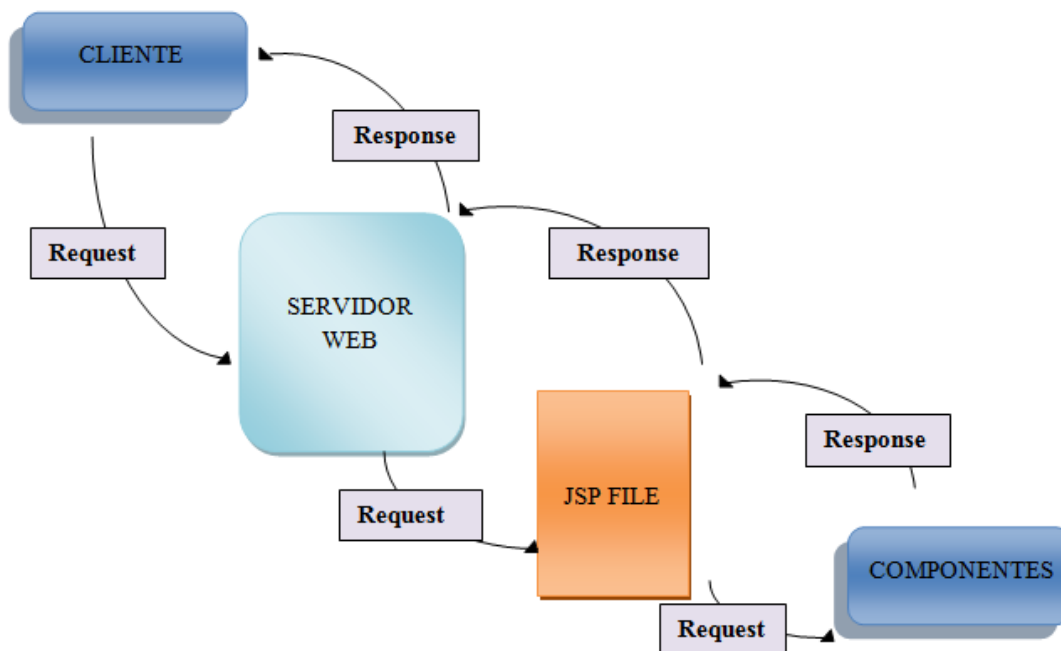
JavaServer Pages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

Sun Microsystems fue una empresa informática recientemente (2009) adquirida por Oracle Corporation, anteriormente parte de Silicon Valley, fabricante de semiconductores y software.

Esta tecnología es un desarrollo de la compañía Sun Microsystems. La Especificación JSP 1.2 fue la primera que se liberó y en la actualidad está disponible la Especificación JSP 2.1.

Las JSP's permiten la utilización de código Java mediante scripts. Además, es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Bibliotecas de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.

3.4.1 Motor JSP



Fuente: Propia

Figura 21: Motor de JSP

El motor de las páginas JSP está basado en los servlets de Java -programas en Java destinados a ejecutarse en el servidor, aunque el número de desarrolladores que pueden afrontar la programación de JSP es mucho mayor, dado que resulta mucho más sencillo aprender que los servlets.

En JSP creamos páginas de manera parecida a como se crean en ASP o PHP otras dos tecnologías de servidor. Generamos archivos con extensión .jsp que incluyen, dentro de la estructura de etiquetas HTML, las sentencias Java a ejecutar en el servidor. Antes de que sean funcionales los archivos, el motor JSP lleva a cabo una fase de traducción de esa página en un servlet, implementado en un archivo class. Esta fase de traducción se lleva a cabo habitualmente cuando se recibe la primera solicitud de la página .jsp,

aunque existe la opción de precompilar en código para evitar ese tiempo de espera la primera vez que un cliente solicita la página.

3.4.2 Arquitectura

JSP puede considerarse como una manera alternativa, y simplificada, de construir servlets. Es por ello que una página JSP puede hacer todo lo que un servlet puede hacer, y viceversa. Cada versión de la especificación de JSP está fuertemente vinculada a una versión en particular de la especificación de servlets.

El funcionamiento general de la tecnología JSP es que el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código Java del servlet a generar. Este servlet será el que genere el documento (típicamente HTML) que se presentará en la pantalla del Navegador del usuario.

El rendimiento de una página JSP es el mismo que tendría el servidor equivalente, ya que el código es compilado como cualquier otra clase Java. A su vez, la máquina virtual compilará dinámicamente a código de máquina las partes de la aplicación que lo requieran. Esto hace que JSP tenga un buen desempeño y sea más eficiente que otras tecnologías web que ejecutan el código de una manera puramente directa.

La principal ventaja de **JSP** frente a otros lenguajes es que el lenguaje Java es un lenguaje de propósito general que excede el mundo web y que es apto para crear clases que manejen lógica de negocio y acceso a datos de una manera prolija. Esto permite separar en niveles las aplicaciones web, dejando la parte encargada de generar el documento HTML en el archivo JSP.

Otra ventaja es que JSP hereda la portabilidad de Java, y es posible ejecutar las aplicaciones en múltiples plataformas sin cambios. Es común incluso que los desarrolladores trabajen en una plataforma y que la aplicación termine siendo ejecutada en otra.

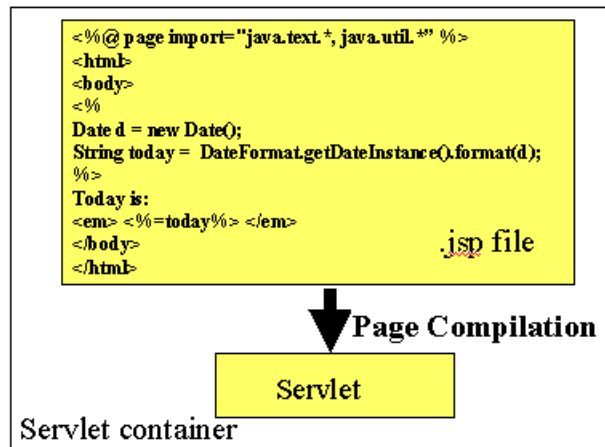
Los servlets y Java Server Pages (JSPs) son dos métodos de creación de páginas web dinámicas en servidor usando el lenguaje Java. En ese sentido son similares a otros métodos o lenguajes tales como el PHP, ASP²³ o los CGIs, programas que generan páginas web en el servidor. Sin embargo, se diferencian de ellos en otras cosas.

Para empezar, los JSPs y servlets se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java para él. Cada servlet (o JSP, a partir de ahora lo usaremos de forma indistinta) se ejecuta en su propia hebra, es decir, en su propio contexto; pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo (cargar programa + intérprete).

Los JSPs son en realidad servlets: un JSP se compila a un programa en Java la primera vez que se invoca, y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un servlet. La principal diferencia entre los servlets y los JSPs es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa Java puro que recibe peticiones y genera a partir de ellas una página web.

A continuación puede ver un ejemplo extremadamente simple de una página JSP y el esquema de conversación de esa página en un servlets.

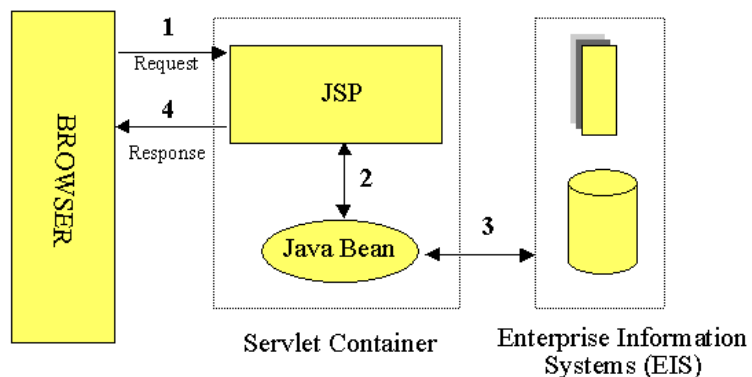
²³ **ASP**. También conocido como **ASP clásico**, es una tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas



Fuente: desarrolloweb.com

Figura 22: Esquema de Conversión de una Página en un Servlets.

3.4.3 Funcionamiento de JSP



Fuente: tejo.usal.es

Figura 23: Descripción del Funcionamiento de una Página JSP

Para poder utilizar esta tecnología es necesario un servidor web que de soporte a paginas HTML y código que implemente un contenedor JSP donde se pueda ejecutar las etiquetas JSP. Existen servidores web que incorporan dicha capacidad dentro de su código, así como servidores escritos íntegramente en Java que dan soporte a esta tecnología directamente.

Para la mayoría de servidores web es necesario añadir código suplementario que implemente el contenedor JSP. Para ello se han desarrollado API's del servidor para poder extender su funcionalidad y dar soporte a JSP.

Una vez que el contenedor JSP ha sido instalado y configurado, los ficheros JSP se tratan igual que los ficheros HTML situándolos en cualquier lugar de la jerarquía de directorios. Cualquier clase Java que se utilice en un fichero jsp debe estar disponible en la variable CLASSPATH del contenedor JSP.

Aunque la especificación JSP no presupone nada sobre la implementación que da soporte a esta tecnología, la mayoría de las implementaciones disponibles están basadas en servlet. El primer componente de las implementaciones basadas en servlets es un servlets especial denominado Compilador de Páginas. Este servlets junto con sus clases Java asociadas se conoce con el nombre de Contenedor JSP. El contenedor está configurado para llamar al compilador de páginas para todas las peticiones que coincida con una página JSP. Su misión es la de compilar cada página JSP en un servlets cuya finalidad es la de generar el contenido dinámico especificado por el documento JSP ORIGINAL.

Para compilar una página el compilador de páginas escanea el documento en busca de etiquetas JSP, generando el código Java correspondiente a cada una de ellas. Las etiquetas estáticas HTML son convertidas a **Strings** de Java una vez que el código del servlets ha sido construido el compilador de páginas llama al compilador de Java para compilar el código fuente y añade el fichero de **bytecodes** resultante al directorio apropiado del contenedor JSP.

Una vez que el servlets correspondiente a la página .jsp ha sido generado, el compilador de páginas invoca al nuevo servlets para generar la respuesta al cliente.

La funcionalidad de una aplicación puede ser integrada de tres modos:

- Como código Java dentro de las paginas JSP, esta no separa la interfaz de la implementación.
- Con el uso de JavaBeans llamados desde las páginas JSP, separa la interfaz de la implementación en gran medida.
- Con el uso de etiquetas personalizadas.

Las páginas JSP pasan por etapas de evolución de su código, las cuales se presentan a continuación:

- Código fuente JSP.- Es el que escribe el desarrollador. Se encuentra en un archivo de texto con extensión .jsp y consiste en una mezcla de código HTML, instrucciones en lenguaje Java, directivas JSP y acciones que describen como generar una página web para dar servicio a una petición concreta.
- Código fuente Java.- El contenedor de JSP traduce el código fuente JSP al código fuente de servlets Java equivalente.
- Clase Java compilada.- Como cualquier otra clase Java, el código de servlets generado se compila en código de bytes en un archivo .class, preparado para ser cargado y ejecutado.

3.4.4 Componentes de una página JSP

Hay tres tipos de elementos JSP:

- Directivas
- Elementos de secuencia de comandos (scripts) que incluyen expresiones scriptles y declaraciones.
- Acciones

3.4.4.1 Directivas

| Atributo | Descripción | Valor por defecto | Ejemplo |
|--------------|-------------------------------------------------------------------------------------------------------|-------------------|---------------------------------|
| info | Descripción de la página. | Cadena vacía. | Info="página de ejemplo" |
| language | Lenguaje de script utilizado. | java | language="java" |
| contentType | Tipo de contenido devuelto por la página. | text/html | contentType="text/html" |
| extends | Superclase de la que se hereda. | Ninguno | extends="paquete.MiClase" |
| import | Importan clases y paquetes para utilizar en la página. | Ninguno | import="java.io" |
| session | Indica si la página pertenece a una sesión. | true | session="true" |
| buffer | Indica si se utiliza un espacio de almacenamiento intermedio antes de enviar la respuesta al cliente. | 8kb | buffer="12kb" |
| autoFlush | Indica el comportamiento que se sigue cuando se ha llenado el búfer. | true | autoFlush="false" |
| isThreadSafe | Indica si la página se puede ejecutar el múltiples hilos de ejecución. | true | isThreadSafe="true" |
| errorPage | Indica la página de error que se debe utilizar si se produce una excepción. | Ninguno | errorPage="dir/paginaError.jsp" |
| isErrorPage | Indica si la página actual es una página de error. | false | isErrorPage="true" |

Fuente: Desarrolloweb.com

Tabla 3.1: Directivas JSP

Las directivas son un conjunto de etiquetas JSP que ofrecen al contenedor de páginas JSP instrucciones específicas de cómo se debe procesar una página determinada. Tiene la forma genérica siguiente:

[% @nombre-de-directiva \[atributo="valor" atributo ="valor..."\] %](#)

3.4.4.2 Elementos de Secuencia de Comandos

| Objetos | Descripción | Clase o interfaz | Categoría |
|-------------|------------------------------------------------------------------------------------------------|----------------------------------------|-------------------------------|
| page | Instancia de la página JSP (servlet) actual. | java.lang.Object | Relacionados con los servlets |
| config | Contiene información relativa a la configuración del servlet generado. | javax.servlet.ServletConfig | Relacionados con los servlets |
| request | Representa la petición realizada a la página JSP. | javax.servlet.http.HttpServletRequest | Entrada/salida |
| response | Representa la respuesta dada por la página JSP. | javax.servlet.http.HttpServletResponse | Entrada/salida |
| out | Representa el flujo de salida del cuerpo de la respuesta HTTP. | javax.servlet.jsp.JspWriter | Entrada/salida |
| session | Permite mantener una sesión para cada uno de los usuarios conectados a nuestra aplicación Web. | javax.servlet.http.HttpSession | Contexto |
| application | Representa a la propia aplicación Web en la que nos encontramos. | javax.servlet.ServletContext | Contexto |
| pageContext | Contexto en el que se ejecuta la página JSP. | javax.servlet.jsp.PageContext | Contexto |
| exception | Excepción que se ha producido en una | java.lang.Throwable | Tratamiento de errores |

Fuente: Desarrolloweb.com

Tabla 3.2: Secuencia de Comandos

Los elementos de secuencia de comandos está compuesta por:

- Declaraciones: Contienen instrucciones de lenguaje Java y se utilizan para definir variables (objetos) y métodos específicos de una página JSP, tanto las variables como los métodos declarados se pueden referenciar por otros elementos de script de la misma página JSP.
- Expresiones. Las expresiones son un medio para obtener acceso al valor de una variable Java u otra expresión y unir ese valor con el HTML.
- Scriptles: Un scriptles es un conjunto de uno o más sentencias o instrucciones en lenguaje Java concebidas para su uso en el procesamiento de una petición http.
- Comentarios: Dentro de una página JSP se puede distinguir tres tipos de comentarios los que son propios de las especificaciones JSP y constituyen un elemento de scripting, los comentarios del lenguaje de scripting, en este caso los comentarios que puede utilizar en el lenguaje Java y los comentarios del lenguaje HTML y XML.
- Objetos implícitos: Aunque los scriptles, las expresiones y los datos de la plantilla HTML están incorporados dentro del método **jspService** (), el contenedor de JSP escribe el esqueleto del propio método, utilizando el contexto de la pagina y un aserie de variables de utilidad. Estas variables están disponibles implícitamente dentro de los scriptles y en las expresiones pero no en las declaraciones.

3.4.4.3 Acciones

Acciones Estándar

Las acciones son elementos JSP de alto nivel que crean, modifican o utilizan otros objetos. A diferencia de las directivas y elementos de secuencia de comandos, las acciones están codificadas usando solamente sintaxis XML.

```
<nombre de etiqueta [ atr =”valor” atr0”valor” ...]>...</nombre de etiqueta>
```

O si la acción no tiene cuerpo, una forma abreviada.

```
<nombre de etiqueta [ atr =”valor” atr=”valor”]/>
```

3.4.5 JAVABEANS

Un bean es simplemente una clase Java que cumple dos requisitos.

- Tiene un constructor de argumento cero
- Implementa `Serializable` o `Externalizable` para hacerlo persistente.

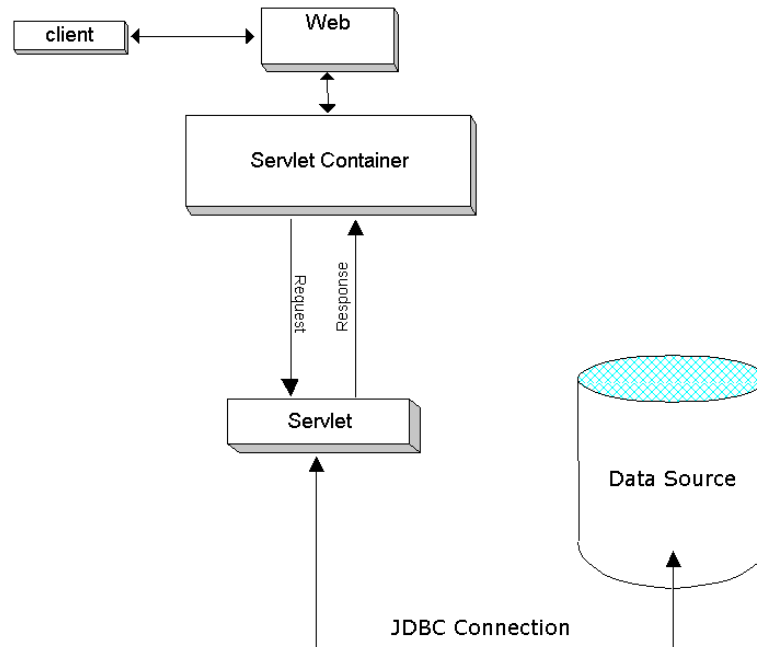
Un bean debe ofrecer alguna forma de hacer que su estado persista durante el tiempo que está activo. Esto se consigue haciendo que el bean implemente la interfaz `Serializable` o la interfaz `Externalizable`.

La socialización hace referencia al proceso de convertir objetos en un flujo de bytes que puede guardarse en un archivo o ser transmitido por una red. El proceso complementario de reunir los objetos de flujo de bytes se llama deserialización.

3.4.6 Etiquetas personalizadas (TAG)

Las etiquetas personalizadas son extensiones similares a XML de la sintaxis y la semántica de JSP que están respaldadas por manejadores de etiquetas creados por el usuario. Las colecciones de etiquetas se organizan en bibliotecas de etiquetas que se pueden empaquetar como archivo JAR, permitiendo su funcionalidad en cualquier motor de servlets adaptado a JSP.

3.4.7 Servlets



Fuente: Oracleapplication.com

Figura 24: Funcionamiento del Servlets

Los servlets son clase Java que amplía la funcionalidad de un servidor web mediante la generación dinámica de páginas web. Un entorno de ejecución denominado motor de servlets administra la carga y descarga de los servlets y trabaja con el servidor web para dirigir peticiones a los servlets y enviar la respuesta a los clientes.

El ciclo de vida de un servlets es como se detalla a continuación:

- El contenedor de servlets crea una instancia de servlets
- El contenedor llama al método **init()** del sevlets
- Si el contenedor tiene una petición para el servlets, se llama al método **service()**
- Después de destruir la instancia el contenedor llama al método **destroy()**
- La instancia es destruida y marcada con una conexión desechada.

Un servlets es construido e inicializado, después se procesan cero o varias peticiones y por último se destruye. En este punto el servlets es totalmente destruido y es una colección de deshechos. El servlets es cargado una sola vez y esta residente en memoria mientras se procesan las respuestas. La interfaz que define esta estructura es **javax.servlet.Servlet**. La Interfaz del servlets define los métodos del ciclo de vida.

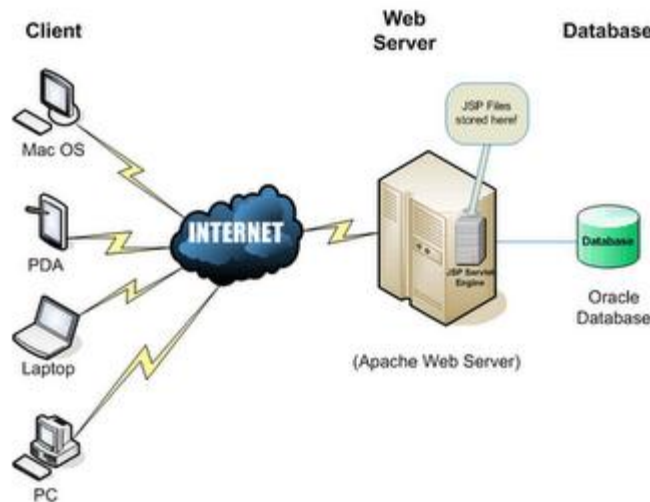
3.4.8 Características de los Servlets

Entre las características principales de los servlets cabe citar las siguientes:

- Los servlets se cargan cuando se los solicita por primera vez y permanecen indefinidamente en la memoria.
- Los servlets se ejecutan en una maquina virtual en un entorno de servidor controlado y solo necesita el HTTP básico para comunicarse con sus clientes.
- Son independientes del servidor utilizado y de su sistema operativo lo que quiere decir que a pesar de estar escritos en Java, el servidor puede estar escrito en cualquier lenguaje de programación.
- Los servlets puede llamar a otros servlets e incluso a métodos concretos de otros servlets en la misma máquina o en la maquina remota. De esta forma se puede destruir de forma más eficiente el trabajo a realizar.
- Los servlets puede obtener fácilmente información acerca del cliente (la permitida por el protocolo HTTP), tal como su dirección IP, el puerto que se utiliza en la llamada el método utilizado (GET, POST), etc. Permite además la utilización de cookies y sesiones de forma que se pueda guardar información específica acerca de un determinado usuario personalizando de esta forma la interacción cliente/servidor. Una clara aplicación es mantener la sección con un cliente.
- Los servlets pueden actuar como enlace entre el cliente y una o varias bases de datos en arquitecturas cliente/servidor. Así mismo puede realizar tareas de

*proxy*²⁴ para un applet debido a las restricciones de seguridad, un applet no puede acceder directamente por ejemplo a un servidor de datos localizados en cualquier maquina remota, pero así podría hacerlo a través de un servlets.

3.5 Acceso a Base de Datos desde JSP.



Fuente: programacionweb.jsp.com

Figura 25: Acceso a la Base de Datos desde JSP

Una aplicación JSP se debe conectar con una base de datos mediante un controlador JDBC. Para conectar mediante un controlador JDBC, deberá especificar algunos valores de parámetros. También se puede utilizar un controlador ODBC si dispone de un controlador puente JDBC- ODBC.

3.5.1 Conectividad a base de datos

Para realizar una conexión hacia la base de datos a través de un JSP o Servlet se utiliza una *driver*²⁵, el cual generalmente es distribuido por la empresa productora de la Base de

²⁴ **Proxy**.-Es aquel que conecta al servidor de acceso a internet de un proveedor de acceso va almacenada toda la información que los usuarios reciban de la web.

Datos, dicho driver toma la forma de un archivo JAR el cual contiene un conjunto de clases necesarias para conectarse a la determinada Base.

Para utilizar un driver en Java basta agregar el archivo JAR al CLASSPATH de ejecución, o bien al directorio /WEB-INF/lib del WAR (“Web-Archive”), esto permite conectarse a la Base de Datos a través del API JDBC ofrecido por el lenguaje Java. Mediante este API es posible buscar, actualizar, eliminar o realizar cualquier otro tipo de operaciones en una Base de Datos a través del lenguaje Java.

Una de las ventajas de aprender a utilizar API JDBC es que un gran gamma de vendedores de Base de Datos lo ha adoptado, esto permite que cualquier código escrito en el sea ejecutable simplemente cambiando el correspondiente driver.

3.5.1.1 Conjunto de Conexiones (“Pool”)

El realizar conexiones de cualquier tipo en un sistema de cómputo es un proceso que implica el uso de diversos recursos, es por esta razón que es recomendable reutilizar cualquier tipo de conexión una vez establecida. La reutilización de estas conexiones establecidas (“Latentes”) se lleva acabo colocándolas en un grupo (“Pool”) para que cualquier programa o recurso del sistema pueda adquirirla, sin incurrir en las penalidades de generar la conexión desde una etapa inicial.

El mantener conexiones hacia Base de Datos en una aplicación Java generalmente es de suma importancia, ya que se realizan búsquedas y actualizaciones constantemente, ante esta constante uso de conexiones se opta por emplear algún tipo de “Pool” hacia Base de Datos.

²⁵ **Driver**.- Controlador que permite conectarse con una base de datos determinada

3.5.2 JDBC

JDBC es un API incluido dentro del lenguaje Java para el acceso a Base de Datos, consiste en un conjunto de clases e interfaces escritos en Java que ofrecen un completo API para la programación de Base de Datos. Debido a que JDBC está escrito completamente en Java, también posee la ventaja de ser independiente de la plataforma. Además, podrá ejecutarse en cualquier sistema que posea una Máquina Virtual de Java.

Básicamente el API JDBC hace posible la realización de las siguientes tareas.

- Establecer una conexión con una base de datos.
- Enviar sentencias SQL
- Manipular los datos
- Procesar los resultados de la ejecución de las sentencia

La columna vertebral de JDBC es el **Driver Manager (gestor de drives)** que se encuentra representado por la clase **java.sql.DriverManager**. El gestor de drives tiene la función primordial de seleccionar el driver adecuado para conectar la aplicación Applet, servlet o página JSP, con una base de datos determinada y acto seguido desaparece.

El API JDBC está formado por una serie de interfaces que permite al programador abrir conexiones con Bases de Datos, ejecutar secuencia SQL sobre las mismas, procesar y modificar los resultados.

3.5.3 Drives o controladores JDBC.

Los drives permiten conectarse con una base de datos determinada. Existen cuatro tipos de drives JDBC a continuación se comenta cada uno de los drives.

Tipo 1.- Puente JDBC-ODBC. Los controladores de este tipo se conectan a bases de datos mediante un controlador ODBC intermedio

Tipo 2.- API nativa, parcialmente en Java. Similar al puente JDBC-ODBC, el controlador de tipo 2 emplea métodos nativos para llamar a las funciones API específicas de la empresa distribuidora.

Tipo 3.- Java puro con software intermedio a base de datos. Estos controladores establecen comunicación mediante un protocolo de red a un servidor de software intermedio, que a su vez establece contacto con uno o más sistemas de administración de base de datos.

Tipo 4.- Java puro directo a base de datos. Los controladores de este tipo llaman directamente al protocolo nativo empleado por el sistema de administración de bases de datos.

La Base de Datos más utilizada con la tecnología JSP es PostgreSQL, PostgreSQL es servidor de Base de Datos relacional libre, liberado bajo la licencia *BSD*²⁶ (Berkeley software distribution). La licencia BSD al contrario que la *GPL*²⁷ permite el uso del código fuente en software no libre.

Algunas de sus principales características son:

Alta concurrencia.

Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla sin necesidad de bloqueos. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

²⁶ **BSD.-** Licencia que permite el uso del código fuente en software libre

²⁷ **GPL.-** Licencia que está orientada principalmente a proteger la libre distribución, modificación y uso del software

Amplia variedad de tipos nativos.

PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IP v4 e P v6)
- Bloques de direcciones estilo CIDR (Interpretación de las direcciones IP)
- Direcciones MAC
- Arrays

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura **GIST** de PostgreSQL.

Funciones.

Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientación a objetos o la programación funciones.

Los disparadores o Triggers son funciones enlazadas a operaciones sobre los datos.

3.6 SEGURIDAD EN JSP

3.6.1 Modelos de Seguridad en JAVA

La seguridad se basa en tres componentes fundamentales del entorno de ejecución:

1.- El cargador de clases, que determina cómo y cuándo pueden cargar código los programas y garantiza que los componentes del sistema no han sido reemplazados.

2.- El verificador de archivos de clases, que garantiza que el código tiene el formato correcto, que el **bytecode** no viola las restricciones de seguridad de tipo de la JVM (Maquina Virtual de Java), que las pilas internas no pueden desbordarse ni por arriba ni por abajo y que las instrucciones en **bytecode** tengan parámetros de tipo correcto. Los **Bytecodes** son en realidad código maquina escrito para el juego de instrucciones de la JVM, por lo que el proceso de interpretación es más rápido que el de otros lenguajes.

3.- El gestor de seguridad, que controla el acceso a los recursos en tiempo de ejecución, como E/S de red y ficheros, creación de cargadores de clases, manipulación de hilos de ejecución, ejecución de programas externos, detener

3.6.1 Dominios protegidos, modelo de permisos y políticas de seguridad.

Dominios protegidos

Conceptualmente un dominio incluye un conjunto de clases cuyas instancias tienen asignados los mismos permisos. Los dominios de protección se determinan mediante la política de seguridad activa en cada momento.

Los dominios protegidos se dividen regularmente en dos categorías.

- Dominios del sistema, que controlan el acceso a los recursos del sistema, tales como, el sistema de archivos, acceso a la red, E/S.
- Dominios de aplicación, que controlan el acceso a los recursos de una aplicación.

Modelo de permisos

Los permisos en Java son clases que representan accesos a recursos del sistema. La clase fundamental es `java.security.Permission`, que es una clase abstracta de la que se deben

definir subclases para representar accesos específicos. Generalmente, una clase de permiso pertenece al paquete en el cual será usada. Por ejemplo, el permiso que representa el acceso al sistema de ficheros local es `java.io.FilePermission`.

Políticas de seguridad

En el *JDK*²⁸ las políticas de seguridad se especifican en uno o más ficheros de configuración de políticas. Estos ficheros especifican que permisos están habilitados para el código obtenido de los orígenes del código especificado.

Un archivo de políticas de seguridad se puede escribir directamente con un editor de texto. Por defecto hay un archivo de políticas del sistema y opcionalmente, otro archivo de políticas de usuario.

3.6.2 Control de acceso a paginas JSP.

Se puede implementar la seguridad en el acceso en nuestras aplicaciones JSP utilizando el objeto sesión.

Las sesiones en las paginas JSP se mantienen entre distintas paginas JSP pertenecientes a una misma aplicación web, además, se puede almacenar información en el objeto sesión en forma de atributos para que se mantenga entre distintas paginas JSP. El objeto sesión no se encuentra disponible en todas las paginas JSP, solo estará disponible en aquellas que pertenezcan a la sesión actual.

²⁸ **JDK** Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en Java.

Laboratorio de
Caranqui

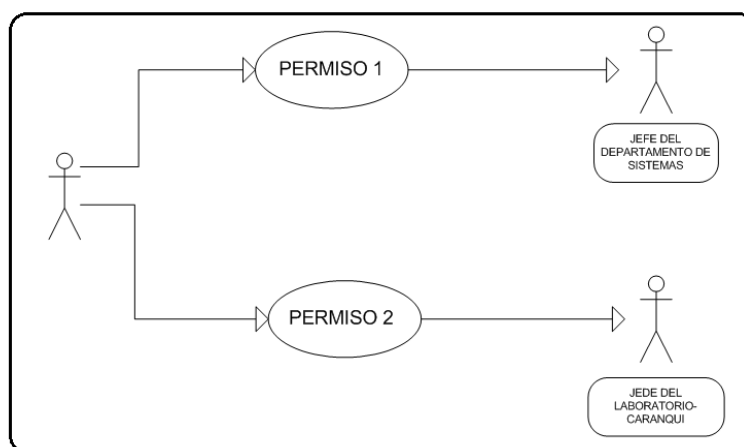
IV
ESPECIFICACIONES
DE CASO DE USO



4. ESPECIFICACIONES

4.1 Especificación de casos de Uso

4.1.1 Ingreso Personal Administrativo



Fuente: Propia

Figura 26: Ingreso Personal Administrativo

Descripción Breve

Este caso de uso describe el proceso de registrar la información de un usuario con cada uno de sus roles y sus permisos otorgados según sus cargos desempeñados.

Flujo básico de Eventos

- ✓ Ingreso de usuarios al sistema con el nombre de usuario y clave. Esto solo lo puede hacer el administrador y los usuarios que están permitidos el manejo del sistema.
- ✓ El sistema verifica que se ingresen usuarios.

- ✓ El sistema verifica que se ingrese *password*²⁹.
- ✓ El sistema verifica confirmación de password.
- ✓ El sistema ingresa a las diferentes aplicaciones permitidas según el rol del usuario.
- ✓ El sistema verifica que se ingresaron los datos en los campos de forma correcta.
- ✓ El sistema almacena toda la información en la base de datos.
- ✓ El sistema confirma el ingreso al usuario.

Flujos alternativos

- ✓ El usuario no completó todos los campos obligatorios.
- ✓ El sistema confirma password no coincide
- ✓ El sistema confirma que faltan datos en el ingreso
- ✓ El usuario se sale del sistema
- ✓ El usuario actualiza el formulario

Pre-condiciones

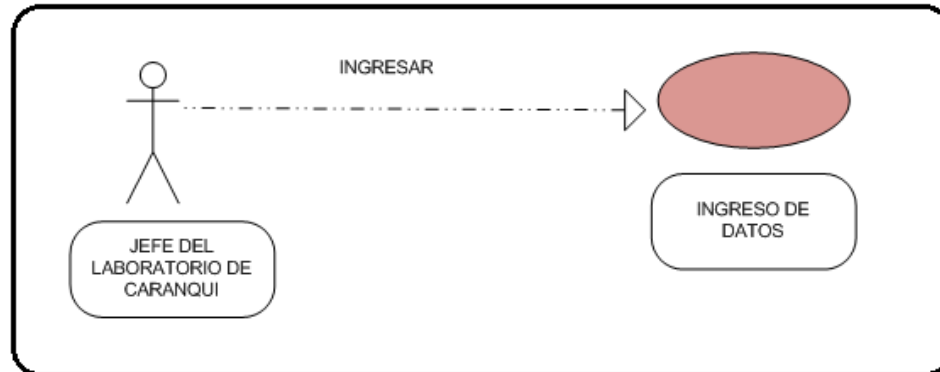
- ✓ Tener todos los datos requeridos del usuario del sistema

Pos-condiciones

- ✓ Datos del empleado quedan almacenados en el sistema.

²⁹ **Password.** Una contraseña o clave (en inglés *password*) es una forma de autenticación que utiliza información secreta para controlar el acceso hacia algún recurso

4.1.2 Ingreso personal Usuarios



Fuente: Propia

Figura 27: Ingreso Personal Usuarios

Descripción Breve

Este caso de uso describe el proceso de registrar la información de todos los usuarios incluidos los administradores del sistema.

Flujo básico de Eventos

- ✓ Ingreso de usuarios al sistema con el nombre de usuario y clave. Esto solo lo puede hacer el administrador y los usuarios que están permitidos el manejo del sistema.
- ✓ El sistema verifica que se ingresen usuarios.
- ✓ El sistema verifica que se ingrese password.
- ✓ El sistema verifica confirmación de password.
- ✓ El sistema ingresa a las diferentes aplicaciones permitidas según el rol del usuario.
- ✓ El sistema verifica que se ingresaron los datos en los campos de forma correcta.
- ✓ El sistema almacena toda la información en la base de datos.
- ✓ El sistema confirma el ingreso al usuario.

Flujos alternativos

- ✓ El usuario no completó todos los campos obligatorios.
- ✓ El sistema confirma password no coincide
- ✓ El sistema confirma que faltan datos en el ingreso
- ✓ El usuario se sale del sistema
- ✓ El usuario actualiza el formulario

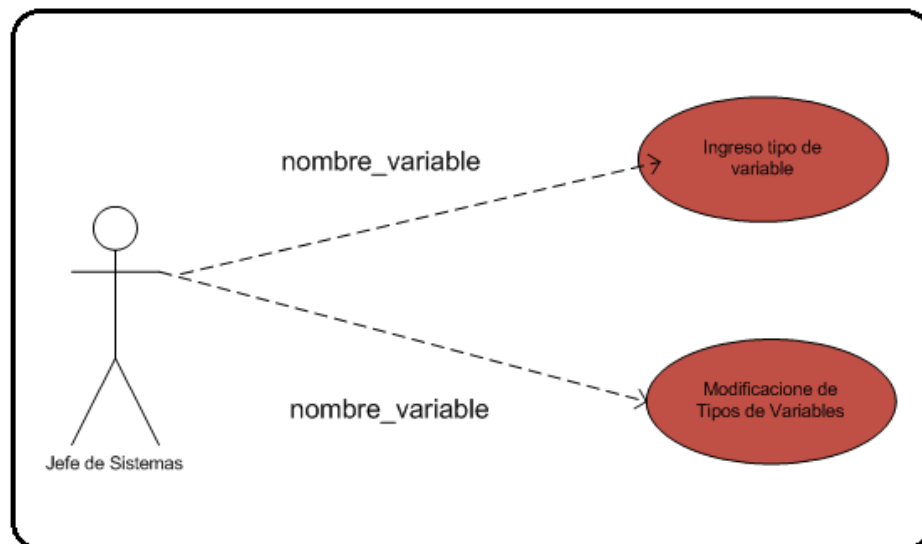
Pre-condiciones

- ✓ Tener todos los datos requeridos del usuario del sistema

Pos-condiciones

- ✓ Datos del empleado quedan almacenados en el sistema.

4.1.3 Ingreso de tipo de variable



Fuente: Propia

Figura 28: Ingreso de Tipo Variable

Descripción Breve

Este caso de uso describe el proceso de ingresar los tipos de variables y las modifica en caso de requerirlas.

Flujo básico de Eventos

- ✓ Ingreso de usuarios al sistema con el nombre de usuario y clave. Esto solo lo puede hacer el administrador y los usuarios que están permitidos el manejo del sistema.
- ✓ El sistema verifica que se ingresen usuarios.
- ✓ El sistema verifica que se ingrese password.
- ✓ El sistema verifica confirmación de password.
- ✓ El sistema ingresa a la pantalla de tipo de variables.
- ✓ El sistema realiza el ingreso de tipos de variables nuevas.
- ✓ El sistema realiza las modificaciones permitidas en el tipo de variables.
- ✓ El sistema almacena toda la información en la base de datos.
- ✓ El sistema confirma el ingreso al usuario.

Flujos alternativos

- ✓ El usuario no completó todos los campos obligatorios.
- ✓ El sistema confirma password no coincide
- ✓ El sistema confirma que faltan datos en el ingreso
- ✓ El usuario se sale del sistema
- ✓ El usuario actualiza el formulario

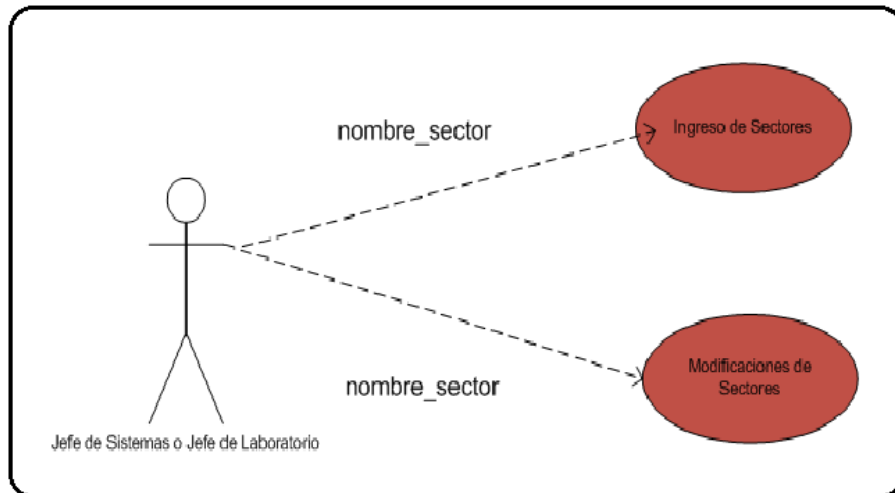
Pre-condiciones

- ✓ Tener todos los datos requeridos del usuario del sistema

Pos-condiciones

- ✓ Datos del empleado quedan almacenados en el sistema.

4.1.4 Ingreso Sectores



Fuente: Propia

Figura 29: Ingreso Sectores

Descripción Breve

Este caso de uso describe el proceso del ingreso y modificación de los sectores que abastece del servicio de agua potable a la ciudad de Ibarra

Flujo básico de Eventos

- ✓ Ingreso de usuarios al sistema con el nombre de usuario y clave. Esto solo lo puede hacer el administrador y los usuarios que están permitidos el manejo del sistema, en este caso las personas encargadas del Laboratorio de Caranqui.
- ✓ El sistema verifica que se ingresen usuarios.
- ✓ El sistema verifica que se ingrese password.
- ✓ El sistema verifica confirmación de password.
- ✓ El sistema ingresa a la pantalla de Ingreso de Sectores.

- ✓ El sistema realiza el ingreso de sectores nuevos.
- ✓ El sistema realiza las modificaciones permitidas en los sectores ya ingresados.
- ✓ El sistema almacena toda la información en la base de datos.

Flujos alternativos

- ✓ El usuario no completó todos los campos obligatorios.
- ✓ El sistema confirma password no coincide
- ✓ El sistema confirma que faltan datos en el ingreso
- ✓ El usuario se sale del sistema
- ✓ El usuario actualiza el formulario

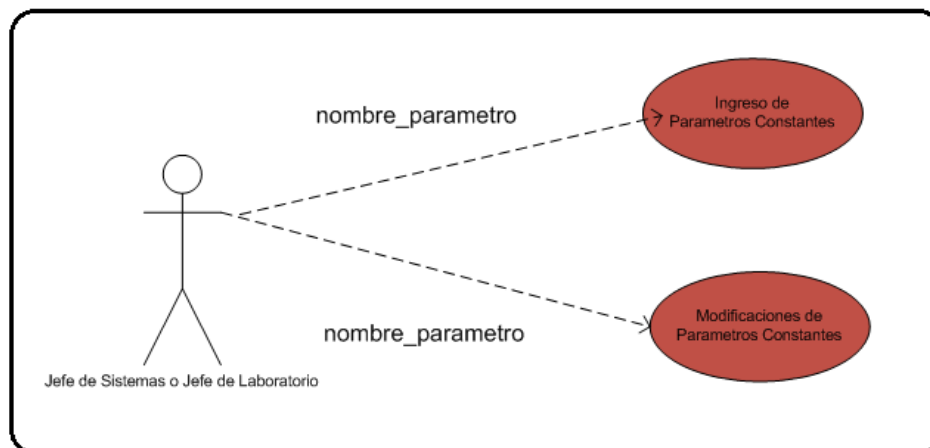
Pre-condiciones

- ✓ Tener todos los datos requeridos del usuario del sistema
- ✓ Tener a disposición la información de los sectores existentes en la ciudad de Ibarra.

Pos-condiciones

- ✓ Datos del empleado quedan almacenados en el sistema.
- ✓ Impresión de los reportes de sectores que dispone el sistema.

4.1.5 Ingreso Parámetros Constantes



Fuente: Propia

Figura 30: Ingreso Parámetros Constantes

Descripción Breve

Este caso de uso describe el proceso del ingreso y modificación de los parámetros que sirven para realizar mediciones en el agua potable como por ejemplo pH, magnesio, etc.

Flujo básico de Eventos

- ✓ Ingreso de usuarios al sistema con el nombre de usuario y clave. Esto solo lo puede hacer el administrador y los usuarios que están permitidos el manejo del sistema, en este caso las personas encargadas del Laboratorio de Caranqui.
- ✓ El sistema verifica que se ingresen usuarios.
- ✓ El sistema verifica que se ingrese password.
- ✓ El sistema verifica confirmación de password.
- ✓ El sistema ingresa a la pantalla de *Parámetros*³⁰ Constantes.

³⁰ **Parámetro.** Un argumento o parámetro es una variable que puede ser recibida por una rutina o subrutina

- ✓ El sistema realiza el ingreso parámetros nuevos.
- ✓ El sistema realiza las modificaciones permitidas en los parámetros ya ingresados.
- ✓ El sistema almacena toda la información en la base de datos.

Flujos alternativos

- ✓ Realiza el ingreso de nuevos parámetros
- ✓ Realiza las modificaciones en los parámetros ya existentes.

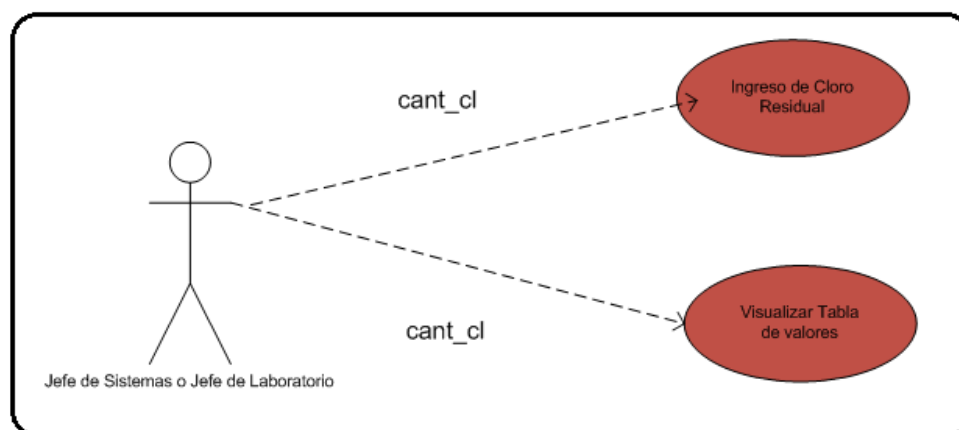
Pre-condiciones

- ✓ Tener todos los datos requeridos del usuario del sistema
- ✓ Tener las listas de parámetros existentes en el Laboratorio

Pos-condiciones

- ✓ Datos del empleado quedan almacenados en el sistema.
- ✓ Imprimir reportes de los parámetros existentes en el sistema
- ✓ Impresión del personal que ha estado manipulando el sistema.

4.1.6 Ingreso Determinación de Cloro Residual.



Fuente: Propia

Figura 31: Ingreso Determinación de Cloro Residual

Descripción Breve

Este caso de uso describe el proceso del ingreso de cloro residual en el agua potable incluyendo los datos del sector, fecha, la hora y la cantidad de cloro.

Flujo básico de Eventos

- ✓ En esta pantalla se realiza el ingreso de los datos específicos para el llenado de los campos.
- ✓ El usuario debe dar un clic en el botón ingresar para guardar los datos en la base de datos.

Flujos alternativos

- ✓ Realizar un clic para regresar a la pantalla anterior
- ✓ El usuario se sale del sistema.

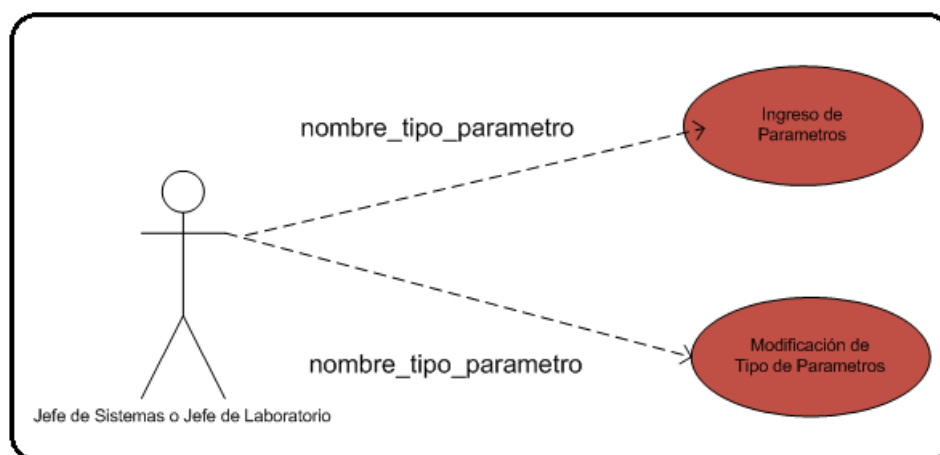
Pre-condiciones

- ✓ El sistema debe de poseer datos en la base de datos para poder observar la grafica.
- ✓ Debe de haberse realizado el ingreso de los sectores con anterioridad a esta pantalla.

Pos-condiciones

- ✓ Imprimir reportes en forma digital.
- ✓ Imprimir estadísticas con el reporte de los datos en tablas.

4.1.7 Ingreso Tipo de Parámetros.



Fuente: Propia

Figura 32: Ingreso Tipo de Parámetros

Descripción Breve

En este caso de uso se realiza el ingreso de los tipos de parámetros fundamentales como son los microbiológicos, análisis físico, análisis químicos y los nuevos que se los ingresara según su necesidad.

Flujo básico de Eventos

- ✓ Se realiza el ingreso de un nuevo parámetro establecido por el Laboratorio de Caranqui.
- ✓ Se realiza un clic para ingresar los nuevos datos a la base de datos del sistema.
- ✓ Se realiza la modificación de acuerdo a la autorización del jefe de Laboratorio

Flujos alternativos

- ✓ Cerrar sesión o salirse del sistema.

- ✓ El cambio de usuario y password.
- ✓ Cuando se realiza una modificación se actualiza los datos en la base datos del sistema.
- ✓ Realizar un clic para regresar al menú del sistema.

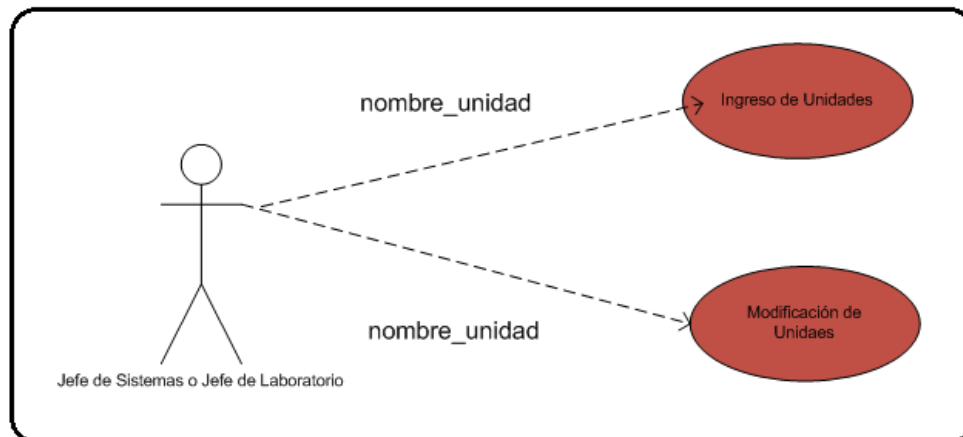
Pre-condiciones

- ✓ Para poder realizar una modificación debe de tener datos anteriores almacenados en la base de datos.

Pos-condiciones

- ✓ Imprimir reportes.

4.1.8 Ingreso Tipo de Unidades.



Fuente: Propia

Figura 33: Ingreso Tipo Unidades

Descripción Breve

El presente sistema nos indica el proceso del ingreso de los nombres de las diferentes unidades dependientes de la empresa de agua potable EMAPA-I, ubicadas en toda la

ciudad de Ibarra así mismo el proceso de las modificaciones de a cuerdo a los requerimientos establecidos por la empresa.

Flujo básico de Eventos

- ✓ Al realizar un clic en el ingreso de las unidades nos aparece que tenemos que llenar los datos del nombre de la unidad con su respectiva descripción.
- ✓ Cuando se realiza una modificación se modifica los campos de la identificación de la unidad, su nombre y su descripción.

Flujos alternativos

- ✓ En la pantalla de actualización de datos puede realizar un clic para regresar a la página de menú.
- ✓ Cerrar sesión salirse del sistema.
- ✓ Modificación de usuario y de password.

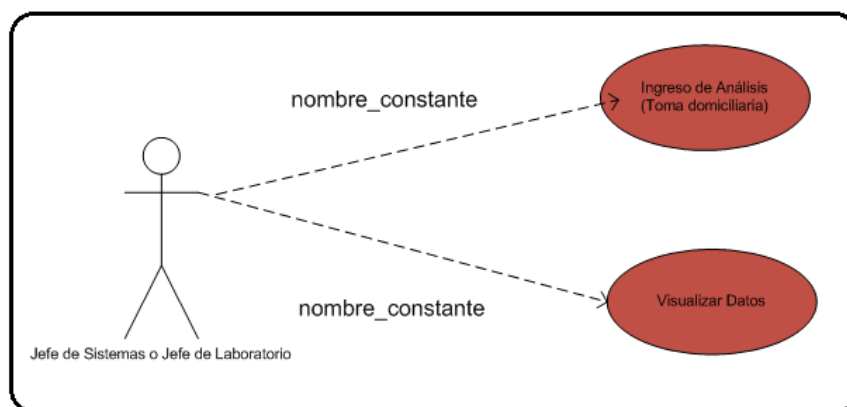
Pre-condiciones

- ✓ Para poder realizar una modificación debe de tener datos anteriores almacenados en la base de datos.

Pos-condiciones

- ✓ Imprimir reportes.

4.1.9 Ingreso de Análisis.



Fuente: Propia

Figura 34: Ingreso de Análisis

Descripción Breve

En el sistema especialmente en este caso de uso nos detalla las tomas de muestras domiciliarias donde los datos de constantes y unidades ya están registrados en el sistema, debemos ingresar sus valores respectivos de cada pantalla que nos presenta el sistema.

Flujo básico de Eventos

- ✓ Al seleccionar el ingreso de análisis nos muestra las selecciones disponibles con las que cuenta el sistema.
- ✓ Realizamos un clic en el botón ingresar para que los datos se almacenen en la base de datos del sistema.
- ✓ Cuando realizamos clic en la opción visualizar nos muestra los datos disponibles del sistema en una tabla.

Flujos alternativos

- ✓ En la pantalla de ingreso realizamos un clic en el botón ingreso y los datos se registran en la base de datos del sistema.

- ✓ En la visualización de las tablas nos indica las fechas de inicio y de finalización que nosotros pretendamos visualizar para obtener los debidos reportes.

Pre-condiciones

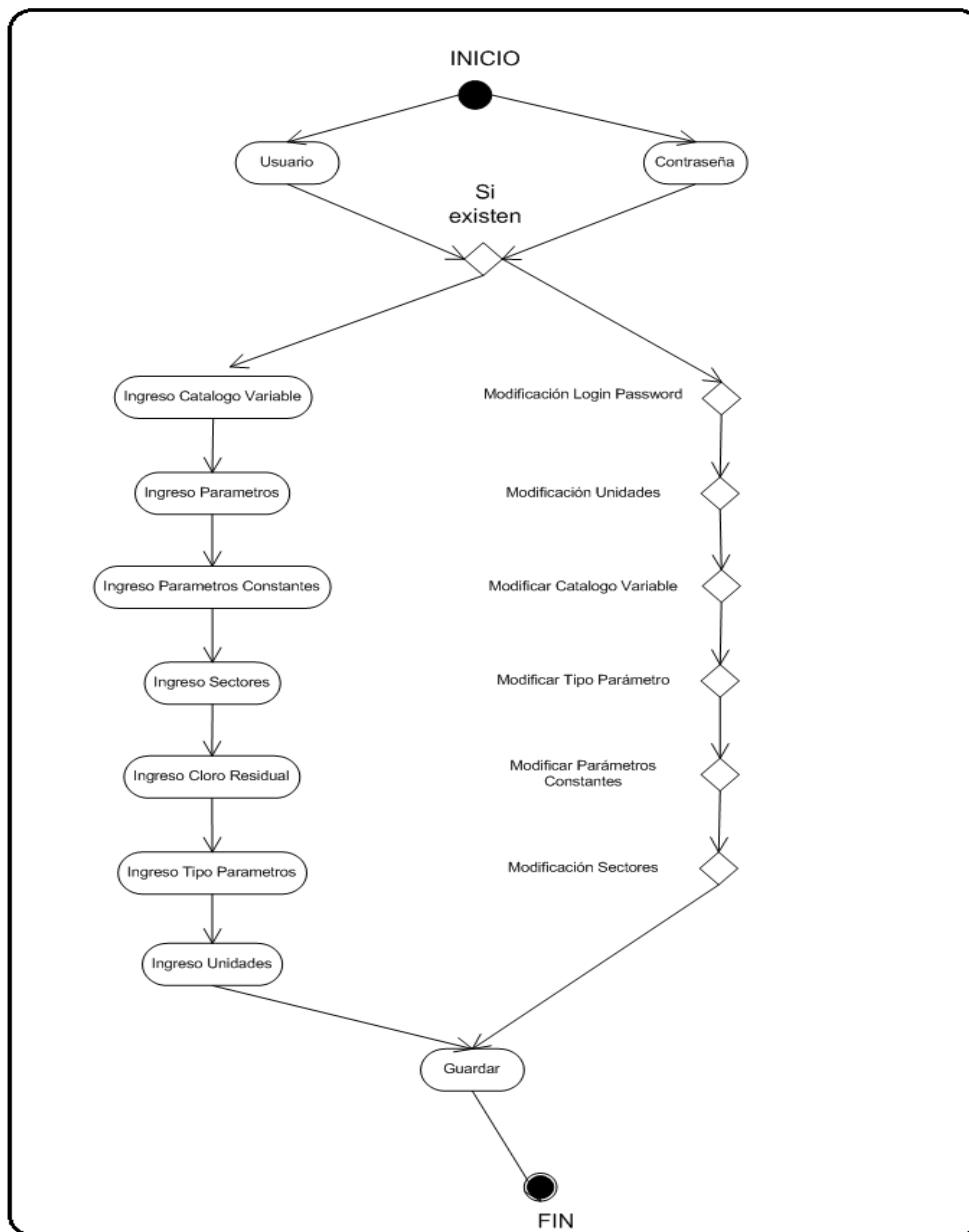
- ✓ En la base de datos tenemos que tener disponibles los campos constantes y unidades.

Pos-condiciones

- ✓ Observar la tabla de toma domiciliaria del Laboratorio de Caranqui.
- ✓ Imprimir reportes.

4.2 Diagrama de Actividades

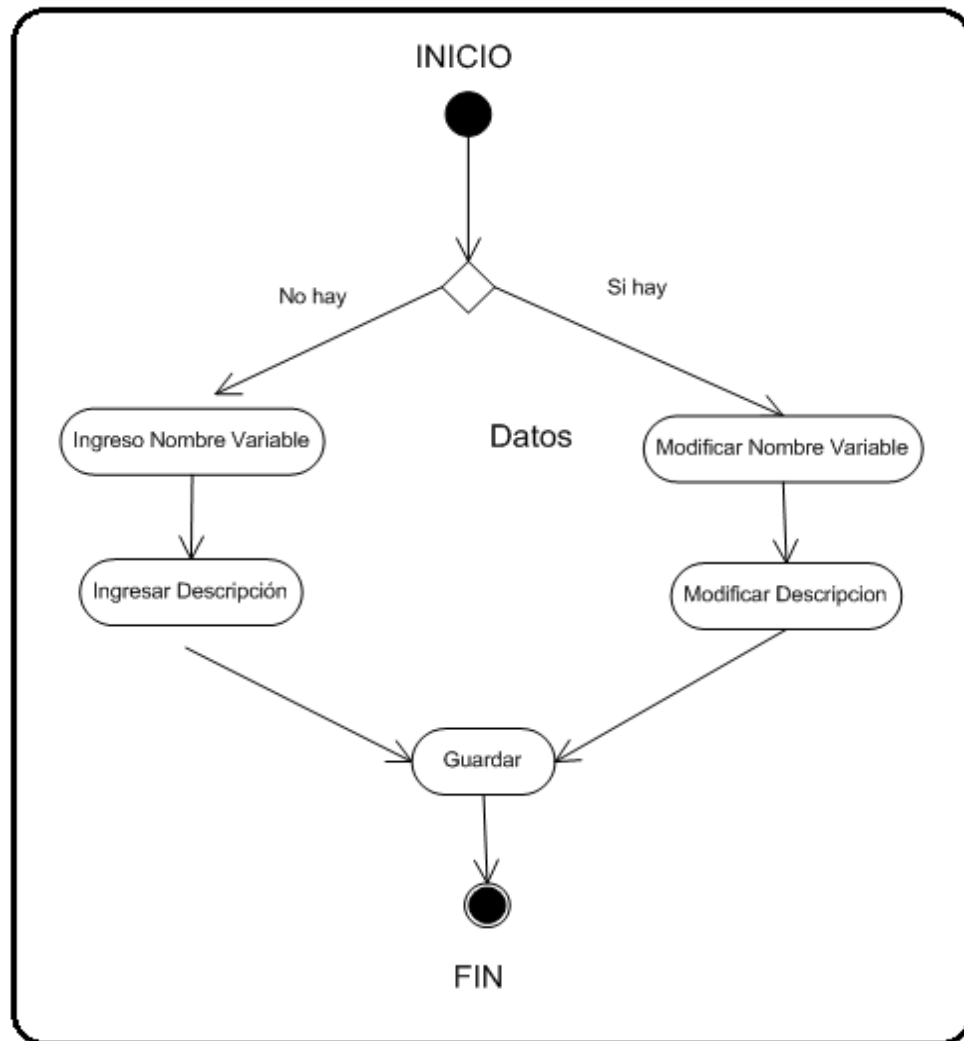
4.2.1 Sistema en General.



Fuente: Propia

Figura 35: Diagrama de Actividaes - Sistema en General

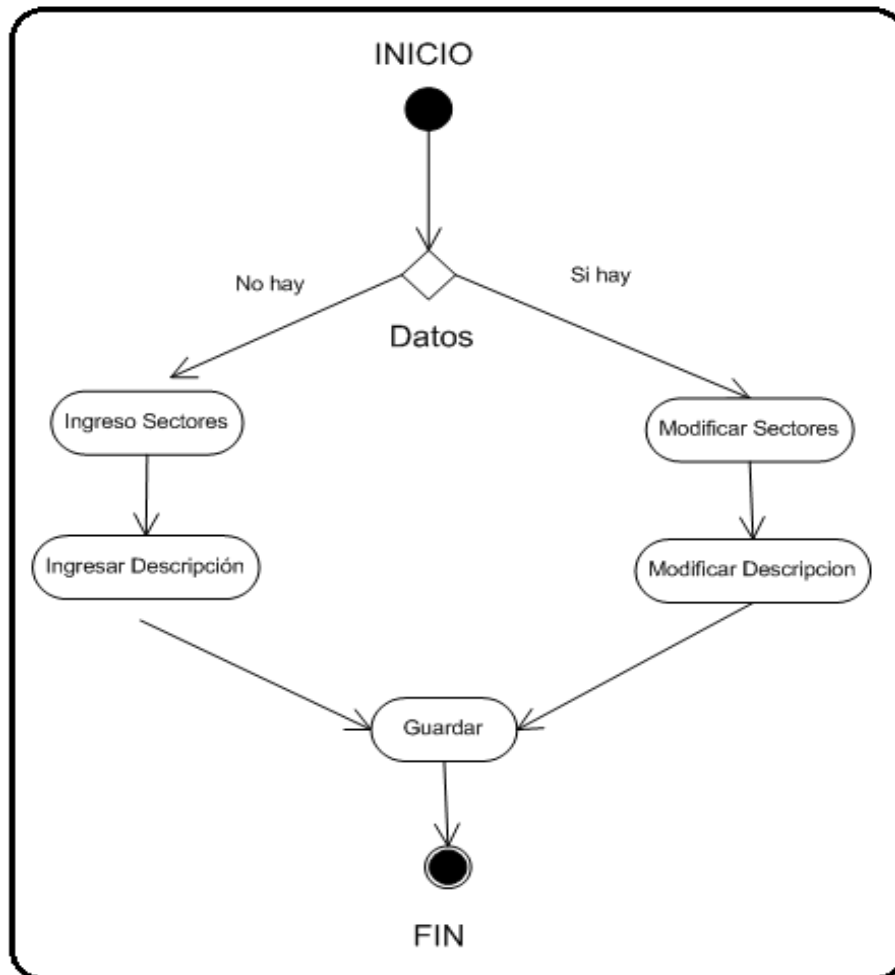
4.2.2 Ingreso Tipo Variable



Fuente: Propia

Figura 36: Diagrama de Actividades – Ingreso Tipo Variable

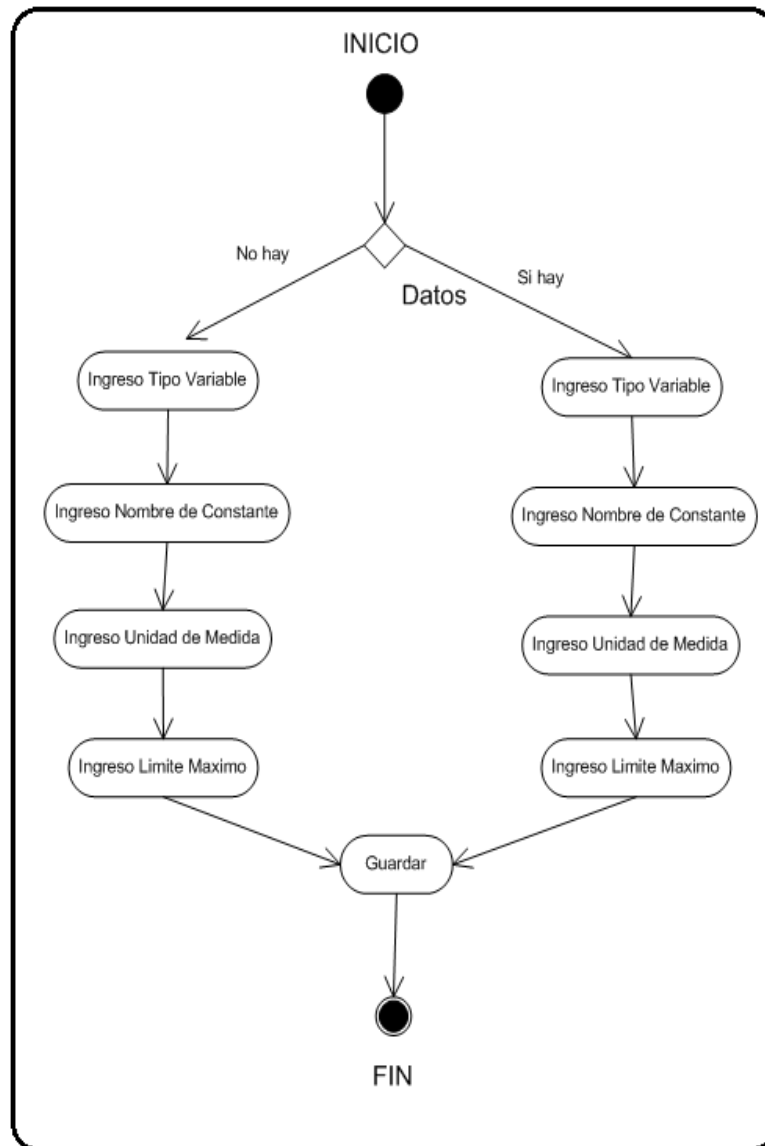
4.2.3 Ingreso Sectores



Fuente: Propia

Figura 37: Diagrama de Actividades – Ingreso Sectores

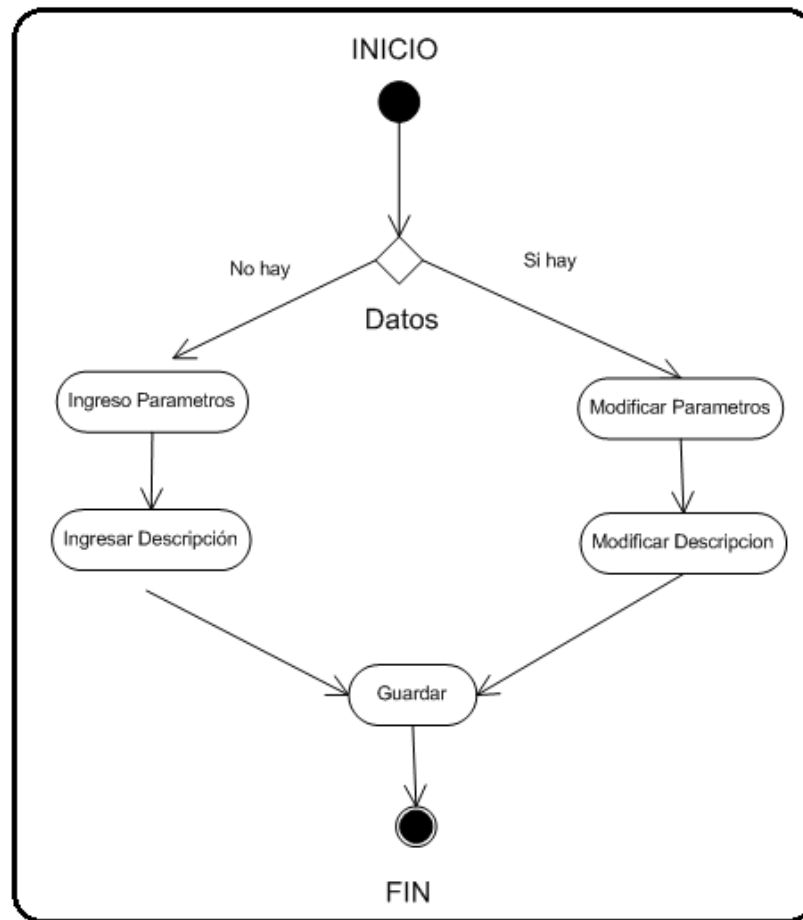
4.2.4 Parámetros Constantes



Fuente: Propia

Figura 38: Diagrama de Actividades – Parámetros Constantes

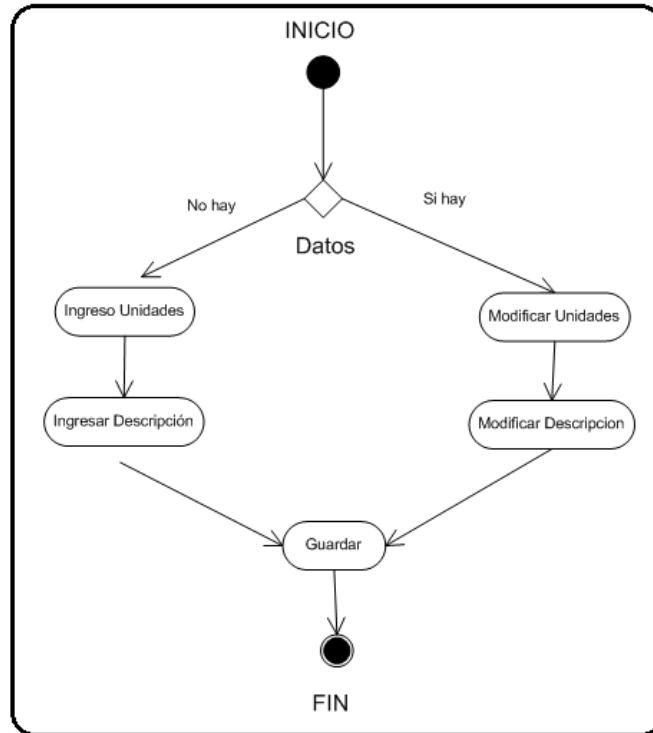
4.2.5 Tipos Parámetros



Fuente: Propia

Figura 39: Diagrama de Actividades – Tipos Parámetros

4.2.6 Ingreso de Unidades

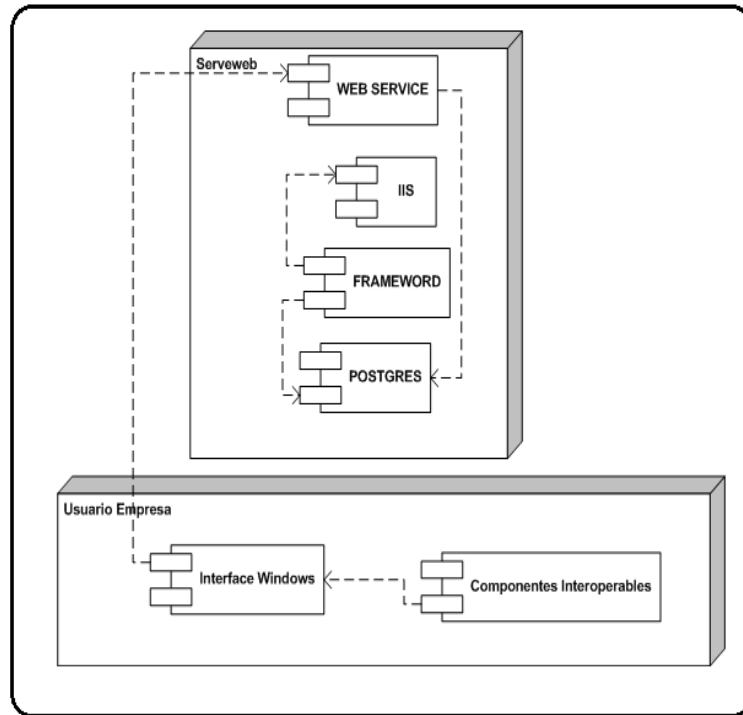


Fuente: Propia

Figura 40: Diagrama de Actividades – Unidades

4.3 Diagrama de Despliegue

Mediante el diagrama de despliegue se puede apreciar el diseño físico de la red y el lugar donde residirán los componentes lógicos desarrollados por la aplicación



Fuente: Propia

Figura 41: Diagrama de Despliegue

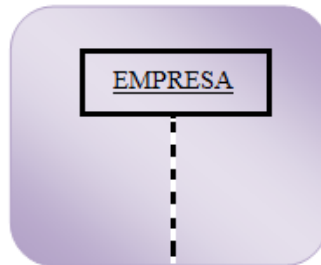
4.4 Diagrama de Secuencia

Luego de obtener el diagrama de proceso de los casos de uso y de despliegue, se procede a representar los estímulos externos y los estados de los objetos o actores que se encontraron en los anteriores diagramas.

El estado de un objeto es un conjunto de valores de las contantes en un cierto momento, el cual cambia al recibir un estímulo llamado evento, por lo que usamos dichos elementos: (objetos y eventos)

Objeto. Es una representación detallada, concreta y particular de un algo, esto determina su identidad, su estado y su comportamiento particular en un instante dado.

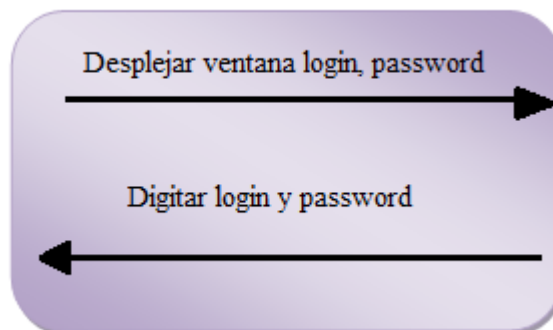
Se lo va a representar mediante un rectángulo con una línea vertical.



Fuente: Propia

Figura 42: Objeto

Eventos. Es toda acción que se da entre los objetos. Se lo va a representar mediante una flecha horizontal que va desde el objeto emisor hasta el objeto receptor.

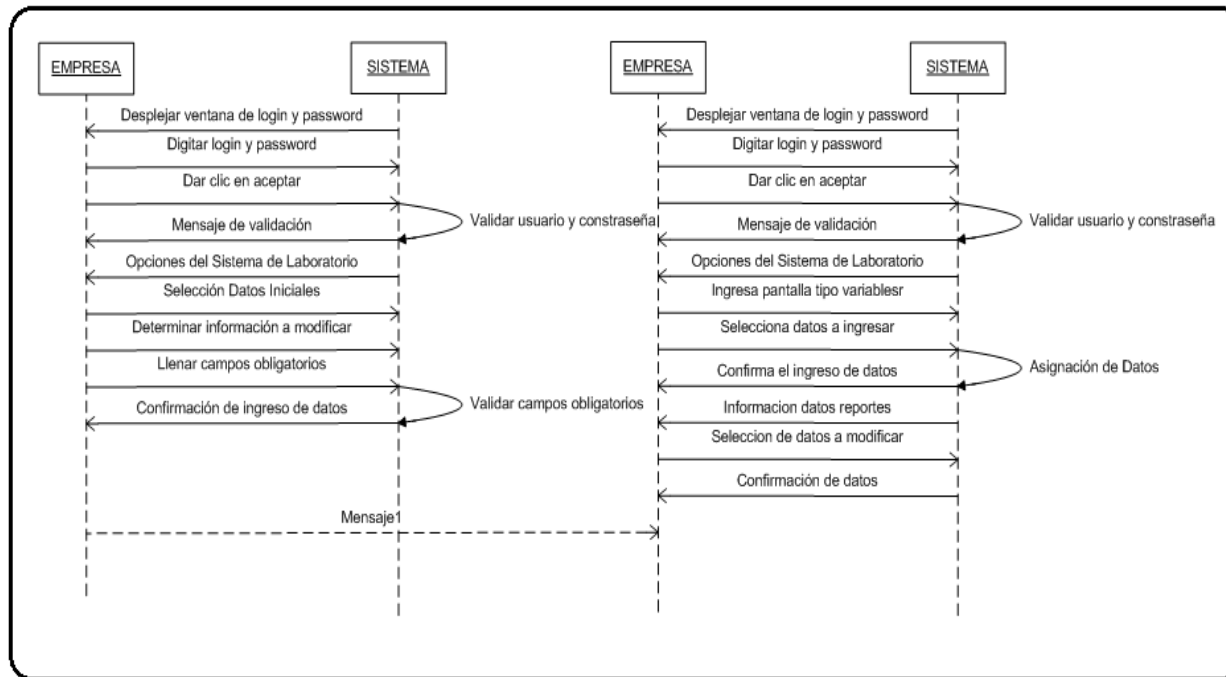


Fuente: Propia

Figura 43: Evento

En los siguientes diagramas se representa las operaciones o flujos de control del modelo funcional, basado en las opciones de usuario que tiene cada actor

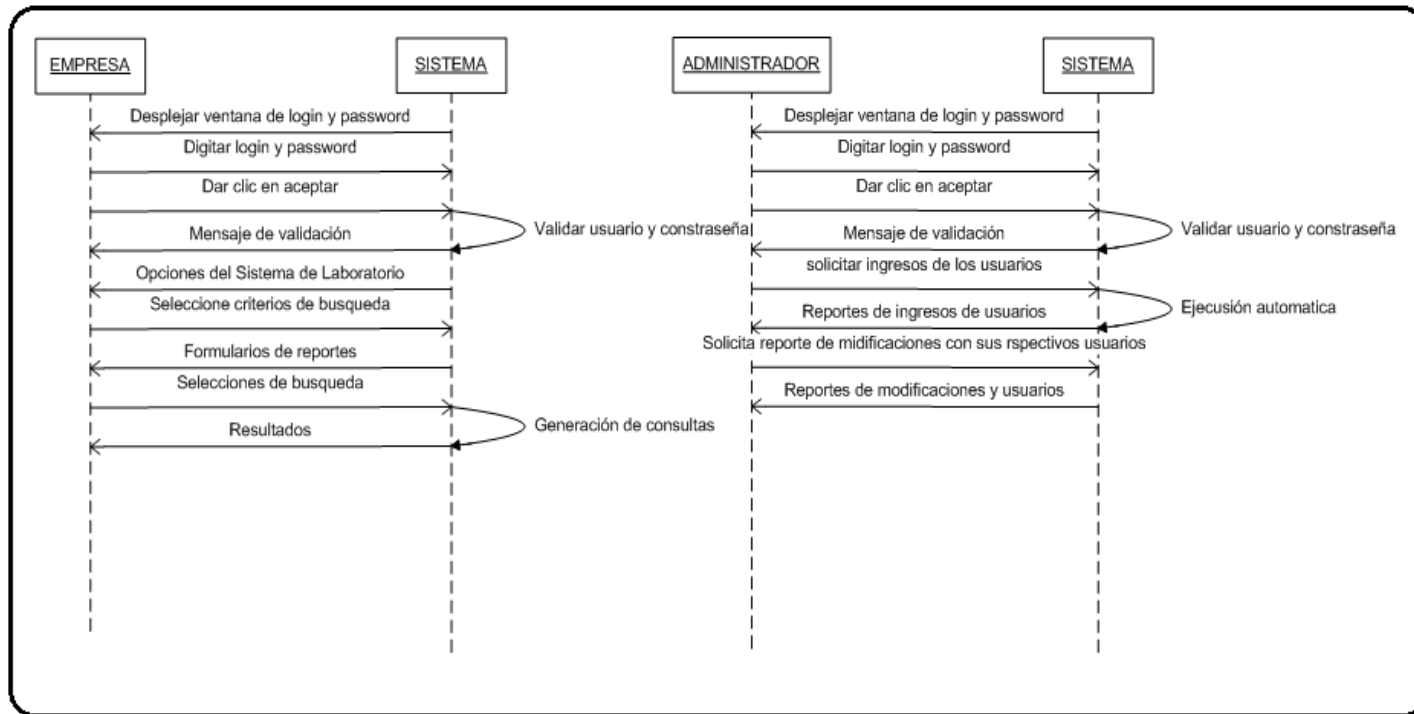
4.4.1 DIAGRAMA DE SECUENCIA: Datos iniciales. Proceso de administración



Fuente: Propia

Figura 44: Diagrama de Secuencia – Datos Iniciales Proceso de Administración

4.4.2 DIAGRAMA DE SECUENCIA: Proceso de Generación de Reportes y Proceso de Reportes de Administración



Fuente: Propia

Figura 45: Diagrama de Secuencia – Proceso de Generación de Reportes y Procesos de Reportes de Administración

Laboratorio de
Caranqui

V
**DOCUMENTACIÓN
TÉCNICA**



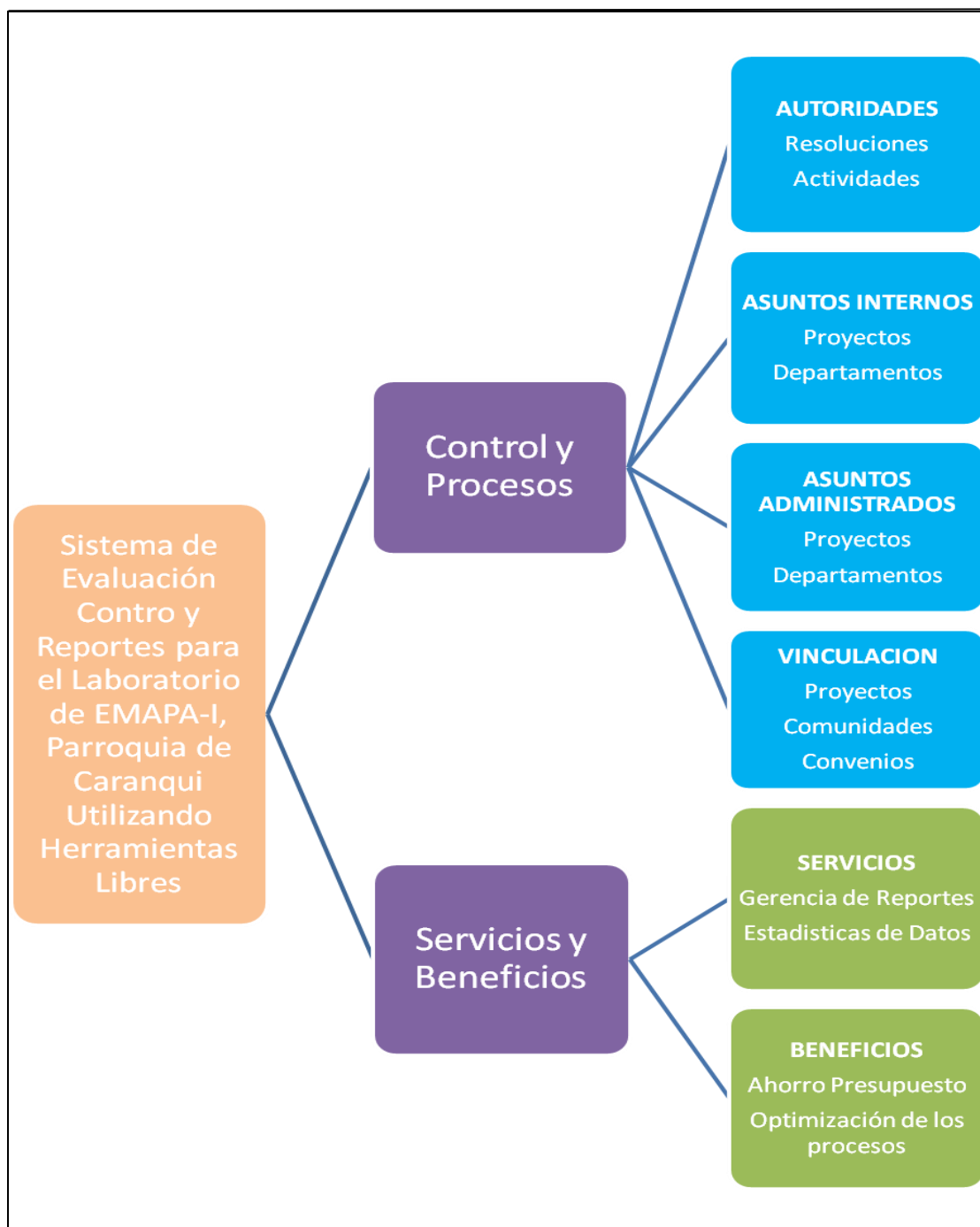
5. DOCUMENTACION TECNICA

5.1 Funcionalidad del Sistema para el Laboratorio de Caranqui

La implementación ha sido desarrollada con el objetivo de contar con un sistema de evaluación, control y reportes que permita una administración adecuada del Laboratorio de Caranqui con el que cuenta EMAPA-I, en lo que respecta al buen funcionamiento del laboratorio y la parte administrativa, permitiendo tener una mejor atención a los abonados con los que cuenta la empresa, optimizando recursos.

El sistema de evaluación, control y reportes brinda los servicios como: controlar y administrar el laboratorio de Caranqui mediante una optimización de recursos a través de la implementación del sistema.

La estructura del sistema de evaluación, control y reportes del Laboratorio está diseñado de tal forma que haya una interacción sin complicaciones para los usuarios finales, en su estructura cuenta con Datos Iniciales, Administración, Reportes, donde se maneja la información de una manera ordenada. Va a estar administrado por un usuario administrador quien es el único que puede modificar, actualizar, eliminar y aumentar información en el sistema de evaluación, control y reportes del Laboratorio de Caranqui.



Fuente: Propia

Figura 46: Funcionalidad del Sistema de Laboratorio

5.1.1 Datos Iniciales, Administración y Reportes

Evaluación y control del Laboratorio de Caranqui

Datos Iniciales: Inicio, pantalla principal del sistema.



Fuente: Propia

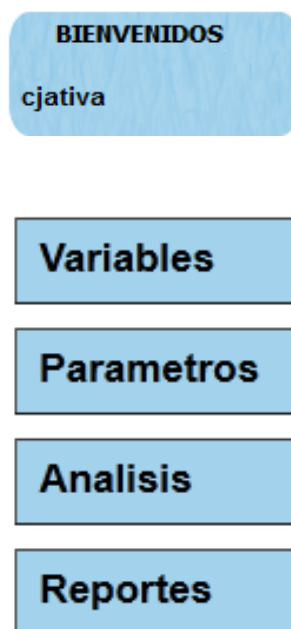
Figura 47: Ingreso de Datos Iniciales en el Sistema

Registrarse: Para poder realizar el proceso de autenticación en el sistema se requerirá de las siguientes claves de acceso:

- **Login:** Es el nombre del usuario el cual se encuentra almacenado en la base de datos.
- **Password:** La clave con que ingresa el usuario que puede ser letras, números o combinados entre las dos opciones.

INGRESAR

Al momento de realizar un clic en el botón ingresar el sistema nos hace una comparación del login y password con los datos que se encuentran almacenados en la base de datos, siempre que la comparación sea exitosa se realiza el ingreso al sistema sin ningún problema.



Fuente: Propia

Figura 48: Menú Principal del Sistema

- **Ingreso tipo Variables:** Realizamos un ingreso de los tipos de variables.
 - Además nos permite el ingreso de datos nuevos en los diferentes campos como son: Nombre y Descripción. De igual manera nos ayuda a modificar los datos ya existentes
- **Ingreso de Sectores:** Se realiza el ingreso de sectores estos son las diferentes ubicaciones en la ciudad de Ibarra que maneja los Laboratorios de Caranqui.
 - Nos permite realizar el ingreso de nuevos datos al sistema como son: Nombre y descripción de los sectores.
 - También realizamos las modificaciones a los datos ya existentes en la base de datos.

- **Parámetros constantes:** Realiza en ingreso de los parámetros constantes una vez llenada la tabla de parámetros ya que contiene un seleccionador de parámetros.
 - Nos permite realizar el ingreso de los diferentes datos que contiene la tabla
 - Modificación de los datos ya existentes, pero una vez seleccionado el parámetro al cual se piensa modificar.

- **Ingreso de datos de determinación de Cloro Residual.** Aquí realizamos los ingresos de la cantidad de cloro residual, pero antes debemos seleccionar el sector en el cual vamos a determinar el cloro residual.
 - Se realiza el ingreso de nuevos datos pero realizando la selección del sector.
 - Una vez ingresado los datos al sistema, se puede realizar las modificaciones pertinentes según el usuario lo requiera.

- **Ingreso Tipo Parámetros.** Aquí realizamos el ingreso de los parámetros y sus respectivas modificaciones si es el caso.
 - En esta pantalla podemos realizar el ingreso de parámetros con su nombre y su respectiva descripción, de aquí es donde parten algunas pantallas de selección.
 - Podemos realizar las respectivas modificaciones de acuerdo a las necesidades requeridas por los usuarios que manipulen el sistema.

- **Ingreso de Unidades.** Para poder realizar el ingreso de las unidades del sistema primeramente tenemos que tener en cuenta que tipo de componente es, por ejemplo: tanques de agua, baños públicos, laboratorios, etc.
 - A través de la pantalla nos permite realizar el ingreso de nuevos datos, que son la id de identificación, el nombre de la unidad con su respectiva descripción.

- Podemos realizar las modificaciones en los diferentes campos que tiene la pantalla.

- **Ingreso Parámetros.** Permite la administración de los parámetros, esto es, su ingreso y su respectiva visualización de los datos en una tabla.
 - Permite realizar el ingreso de las observaciones que tengan los diferentes parámetros, este procedimiento lo podemos realizar una vez seleccionada las opciones de tipo de parámetro y tipo de constante
 - Podemos visualizar los datos una vez seleccionado los parámetros que deseemos ver.

- **Ingreso de Análisis.** Permite realizar los nuevos ingresos de análisis y la visualización de los datos ya registrados en la base de datos
 - Lo primordial aquí es ingresar el valor del análisis y la fecha que se ha realizado este proceso, una vez seleccionado las opciones de nombre de constante y unidad.
 - Podemos obtener los reportes que genera el sistema seleccionando las fechas de inicio y fin.

- **Web Services:** es la parte donde enlace los datos desde un servidor remoto hasta el cliente que puede visualizar la información solicitada de acuerdo a lo permitido por el sistema del cliente, este cuenta con algunos métodos los mismos que constan de la siguiente manera:
 - **Ingreso y Actualización de Datos:** este método nos permite el ingreso de información desde el cliente hasta la base de datos, al igual que nos permite una actualización de los mismos.
 - **Obtener Datos:** este método permite obtener la información necesaria para que el cliente pueda visualizarla.

5.1.2 Tareas del Administrador

Para administrar y mantener el sistema de evaluación, control y reportes del Laboratorio de Caranqui, funcional y actualizado el administrador tiene las siguientes funciones y responsabilidades:

- **Desarrollar la programación:** el administrador debe programar cualquier solución que se presente en el Sistema Evaluación, Control y Reportes del Laboratorio de Caranqui.

- **Dar mantenimiento al Web services:** el administrador debe darle el seguimiento al mismo, debe observar que esté funcionando correctamente, que siempre este revisando los datos que son enviados de la parte del usuario en tiempo en que se realizan las debidas mediciones y el ingreso de datos nuevos al sistema; que responda a tiempo las peticiones realizadas para poder realizar los reportes visuales o los reportes impresos; que se ayude a la persona encargada de manejar el sistema en cualquier problema técnico que tenga para realizar el monitoreo del sistema del Laboratorio de Caranqui.

- **Administrador de la base de datos:** Es la persona responsable de los aspectos ambientales de una base de datos. En general esto incluye aspectos como la **recuperabilidad** en la cual el administrador crea y prueba resultados; **integridad** donde se ayuda a verificar los datos; **seguridad** con lo cual se implementa controles de acceso a los datos; **disponibilidad**, es decir se asegura mayor tiempo de encendido; **desempeño** lo que asegura el máximo rendimiento incluso con las limitaciones; **desarrollo y soporte a pruebas** mediante esta acción se ayudar a los programadores e ingenieros a utilizar eficientemente la base de datos.

- **Chequeo de errores:** En lo que se refiere a la manipulación del computador siempre se encuentra con errores ya sean estos de digitación o propios del

sistema, la tarea que el sistema realiza al hacer un chequeo de errores es la comparación de los datos ingresados por los usuarios y los datos que se encuentran almacenados en la base de datos. Por ejemplo la autenticación del usuario con su respectiva contraseña no siempre se ingresa la palabra correcta y el sistema imprime un mensaje de error para que el usuario sepa donde existe el problema y tratar de corregirlo.

Laboratorio de Caranqui

VI CONCLUSIONES RECOMENDACIONES



6. CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- Implementar el Sistema Evaluación, Control y Reportes del Laboratorio, generará un cambio en el proceso de registro y almacenamiento de la información mediante su tecnificación, atendiendo todos los requerimientos de información en tiempo real para los usuarios del sistema propuesto
- Optimizar las tareas y los reportes diarios que se envían a la gerencia para su revisión por parte del Administrador del laboratorio, ahora se generarán automáticamente en el sistema y la gerencia obtendrá los reportes para su aprobación desde cualquier terminal.
- La base de datos PostgreSQL implementada por requerimiento de la Institución para la elaboración del sistema propuesto, cumple con los requerimientos para el almacenamiento masivo de información, con una ventaja complementaria que es Open Source.
- Los reportes del sistema se han estandarizado según los requerimientos de las áreas, para lo cual se utilizó la metodología RUP. Esto implica conocer el proceso que ejecutan en el laboratorio y su apropiada documentación, requerimientos de las áreas, obteniendo como resultado final un proceso optimizado. Complementariamente queda un registro detallado de todos los cambios realizados en el transcurso hasta la finalización del mismo.
- Mediante la implementación de este sistema colaboramos al cuidado del medio ambiente al reducir notablemente la utilización de papel y al mismo tiempo se está dando cumplimiento con lo estipulado en la Constitución de la Republica del Ecuador que dice en su Capítulo II sobre Biodiversidad y Recursos Naturales en

su Sección Primera: Naturaleza y Ambiente, en sus Artículos 395 y 396. En donde la constitución reconoce algunos principios ambientales.

- Se ha programado el sistema logrando la fiabilidad del mismo fortaleciendo la información que se almacenara en la base de datos además con las respectivas seguridades que brinda el administrador.
- La programación en capas ha sido una de las últimas tendencias en cuanto a software comercial se refiere, es una tendencia que bien aplicada puede resultar en un desarrollo de software eficiente.

6.2 RECOMENDACIONES.

- Al realizar la implementación del Sistema de Evaluación, Control y Reportes del Laboratorio de Caranqui, se debe tener en consideración las siguientes recomendaciones que ayudará al buen funcionamiento y desempeño del Sistema:
- Capacitación en la utilización de paquetes básicos de computación al personal que trabaja en el Laboratorio para lograr una optima operatividad del presente sistema.
- Implementar equipo de última generación para lograr un tratamiento eficaz del agua y por ende brindar un mejor servicio a la comunidad de la ciudad de Ibarra.
- Realizar evaluaciones periódicas a cada una de las áreas por parte del administrador del sistema

- Que la administración del sistema se encargue directamente al departamento de Informática de la empresa de agua potable EMAPA-I

Laboratorio de
Caranqui

VII
GLOSARIO DE
TERMINOS.



7. GLOSARIO

- **Open Source** Es el software que está licenciado de tal manera que los usuarios pueden estudiar, modificar y mejorar su diseño mediante la disponibilidad de su código fuente.
- **Pymes** Pequeñas y medianas empresas.
- **HTTP** Protocolo de transferencia de hipertexto es el método más común de intercambio de información en la world wide web, el método mediante el cual se transfieren las páginas web a un ordenador.
- **NCSA** Centro Nacional de Aplicaciones de Supercomputadores
- **Servidor Apache** Un servidor de páginas web, es un programa que permite acceder a páginas web alojadas en un ordenador.
- **RUP** Rational Unifiel Process.
- **Artefactos** Elementos materiales que los humanos han construido o modificado
- **CGI** Por sus siglas en inglés ‘Common Gateway Interface’ es de las primeras formas de programación web dinámica.
- **Stakeholder** Cualquier persona interesada en, afectada por y/o implicada con el funcionamiento del sistema o software
- **JSP** Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo
- **JAVA** Es un lenguaje de programación con el que podemos realizar cualquier tipo de programa
- **Script** En informática es un guion, archivo de órdenes o archivo de procesamiento por lotes
- **HTML** Las siglas de HyperText Markup Language («lenguaje de marcado de hipertexto»), es el lenguaje de marcado

- predominante para la elaboración de páginas web.
- API Una interfaz de programación de programa y aplicaciones. Conjunto de normas que determinan como debe usarse una determinada función de un programa en una aplicación.
 - Applets Java Es un applet escrito en el lenguaje de programación Java, pueden ejecutarse en un navegador web.
 - Servlets Aplicación sin interfaz gráfica que se ejecuta en un servidor de Internet, procesando información HTML previamente recogida por un navegador web.
 - JVM Maquina virtual de Java.
 - XML Lenguaje de Marcas Extensible.
 - cookies Son pequeños archivos de texto que son descargados automáticamente, al navegador en una página Web específica. En una cookie se almacena cierta información sobre el visitante que la página considera importante recordar. Una cookie se usa por ejemplo para que cada vez que accedamos a una página esté adaptado a nuestro gusto. También sirve para la persistencia de secciones
 - Sesiones Es la secuencia de páginas que un usuario visita en un sitio web.
 - Service Este método es la parte medular de todo Servlet ya que dentro de él se incluyen las tareas principales de ejecución.
 - Destroy Se llama cuando ya no se va a utilizar más el applet, cuando se necesita que sean liberados todos los recursos dispuestos por el applet,
 - NetBeans Es un modelo de componentes creado por Sun Microsystems para la construcción de aplicaciones web

- SGBD Sistema de Gestión de Base de Datos
- Proxy Un servidor proxy actúa como una barrera o servicio de seguridad entre la intranet e internet, lo que evita que otras personas obtengan acceso en Internet a información confidencial en la red interna o en el equipo. El administrador del sistema de telefonía puede proporcionarle el nombre o la dirección IP correctos para especificarlos aquí.
- Driver Controlador que permite conectarse con una base de datos determinada
- TCP/IP Protocolo de control de transmisiones / Protocolo Internet. Es el protocolo estándar de comunicaciones en red utilizado para conectar sistemas informáticos a través de Internet
- bytecodes Es un código intermedio más abstracto que el código máquina. Habitualmente es tratado como un fichero binario que contiene un programa ejecutable similar a un módulo objeto, que es un fichero binario producido por el compilador cuyo contenido es el código objeto o código máquina .
- JDBC Se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de base de datos
- ODBC (Open Database Connectivity) es un programa de interface de aplicaciones (API) para acceder a datos en sistemas manejadores de bases de datos tanto relacionales como no relacional
- PHP Es un lenguaje de programación interpretado, diseñado

originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica

- ASP También conocido como **ASP clásico**, es una tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente
- WAR Driver Manager (Gestor de Drives)

Laboratorio de
Caranqui

VIII
REFERENCIAS
BIBLIOGRAFICA



8. REFERENCIAS BIBLIOGRAFICAS

8.1 REFERENCIAS LIBROS

- [LIB .01].- Grupo EIDOS, Tecnologías de Servidor con Java. Servlets, JavaBeans, JSP
- [LIB .02].- Deitel. Java 5ta ed. DOO con UML y los patrones de diseño JDBC, SERVLET, JSP
- [LIB .03].- Mc Graw – Hill, Manual de referencia JSP.
- [LIB .04].- Paul Tremblett, Superutilidades para Java Server Pages.
- [LIB .05].- Jayson FALKNER, Ben Galbraith, Sathya Nayrayana, John Timey, Fundamentos de desarrollo web con JSP.
- [LIB .06].- Richard MANSON Haefel, Enterprise Java Beans.
- [LIB .07].- Constitución de la Republica del Ecuador 2008
- [LIB .08].- Ingeniería del Software, Un enfoque práctico. Sexta Edición. Roger S Presuman. 2005.
- [LIB .09].- Servlets y Programación JSP, Juan Antonio Palos, 2004
- [LIB .10].- Profesional Apache Tomcat 6, Vivet Chopra, Anaya Multimedia, 2008
- [LIB .11].- Ingeniería de Software, Lan Sommerville, 2005
- [LIB .12].- Como Programar en JAVA Harvey M Deitel, Paul J. Deitel, Guillermo Trujano Mendoza, 2004
- [LIB .13].- Introducción XHTML, Javier Eghíluz Pérez, 2004
- [LIB .14].- XML Conceptos y Implementación, Johnny Brochard, 2001

8.2 REFERENCIAS WEB

- [WWW.01].-<http://www.desarrolloweb.com/articulos/831.php>
- [WWW.02].-<http://es.wikipedia.org/wiki/JSP>
- [WWW.03].-<http://java.sun.com/products/jsp>
- [WWW.04].-<http://javaboutique.internet.com/tutorials/JSP>
- [WWW.05].- www.apl.jhu.edu/~hall/java/Servlet-Tutorial/servlet-Tutorial-JSP.html
- [WWW.06].-<http://www.sun.com/software/jwebserver>
- [WWW.07].-<http://jakarta.apache.org>
- [WWW.08].-<http://www.klomp.org/gnujsp>
- [WWW.09]. <http://www.scribd.com/doc/395783/RUP-etapa-diseno>
- [WWW.10].<http://es.wikipedia.org/wiki/PostgreSQL>
- [WWW.11].<http://www.postgresql.org>
- [WWW.12]. <http://lawebdelprogramador.com/>
- [WWW.13].http://www.programacion.net/java/tutorial/servlet_jsp/
- [WWW.14].http://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas, 1 Junio 2007.
- [WWW.15].<http://phpsenior.blogspot.com/2006/07/conceptos-separar-el-codigo-de-la-capa.html>, 2006.
- [WWW.16].http://www.informatizate.net/articulos/el_acceso_a_datos_en_asp_net_whidbey_parte_02_20050219.html
- [WWW.17].<http://sentidoweb.com/2006/12/26/spring-framework-de-java.php>, 26 diciembre 2006.
- [WWW.18].http://www.programacion.com/java/articulo/jap_persis_hib, 2007.

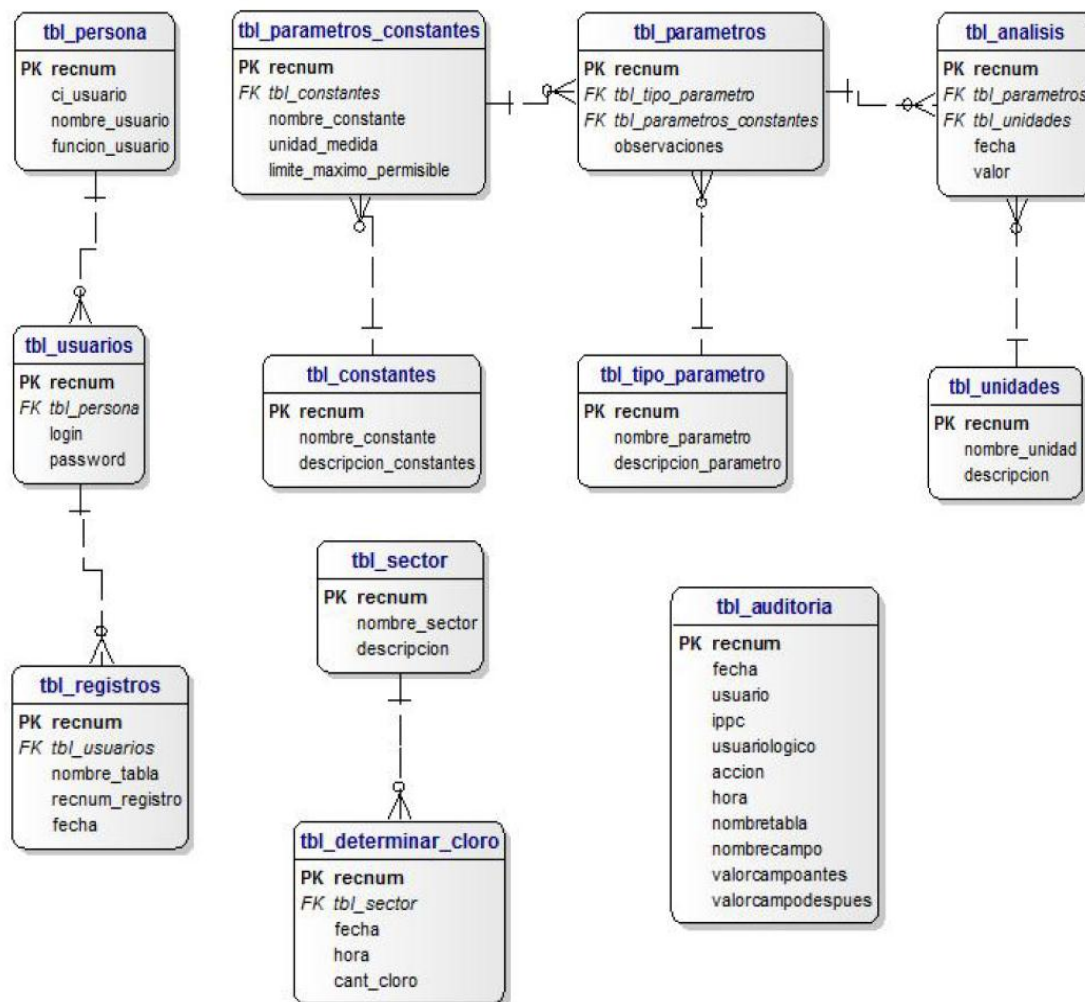
Laboratorio de
Caranqui

X
ANEXOS



9. ANEXOS

9.1 ANEXO A : BASE DE DATOS



Fuente. Propia

Figura 49: Base de Datos del Sistema

9.2 ANEXO B: DICCIONARIO DE DATOS

Tabla: tbl_analisis.

Descripción: Contiene los datos de los análisis dependiendo de los parámetros.

| Campo | Tipo | Nulo | Predeterminado | Descripción | P. key | F.key |
|----------------|-----------|------|----------------|----------------------------------------------------------|--------|-------|
| regnum | bigserial | No | Null | Generación automática de número para cada nuevo registro | | |
| tbl_parametros | bigserial | No | Null | Este campo almacena los parámetros | x | |
| tbl_unidades | bigserial | No | Null | Este campo almacena la unidades | | x |
| fecha | date | No | | Guarda la fecha | | |
| valor | real | No | | Guarda el valor | | |

Fuente: Propia

Tabla A.1: Tabla de Análisis

Tabla: tbl_constantes.

Descripción: Contiene los datos de los análisis dependiendo de los parámetros.

| Campo | Tipo | Nulo | Predeterminado | Descripción | P.Key | F.key |
|------------------------|-------------------|------|----------------|--------------------------------------------|-------|-------|
| regnum | bigserial | No | Null | Generación automática de un nuevo registro | x | |
| nombre_constante | character varying | No | 100 | Almacena el nombre de la constante | | |
| descripcion_constantes | text | No | | Almacena la descripción de la constante | | |

Fuente: Propia

Tabla A.2: Tabla de Constantes

Tabla: tbl_determinar_cloro.

Descripción: Contiene los valores del cloro residual.

| Campo | Tipo | Nulo | Predeterminado | Descripción | P.Key | F.key |
|------------|-------------------|------|----------------|--------------------------------------------|-------|-------|
| regnum | bigserial | No | Null | Generación automática de un nuevo registro | x | |
| tbl_sector | bigserial | No | Null | Almacena el nombre del sector | | x |
| fecha | date | No | | Almacena la Fecha | | |
| hora | character varying | No | 20 | Almacena la Hora | | |
| cant_cloro | character varying | No | 40 | Almacena la cantidad | | |

Fuente: Propia

Tabla A.3: Tabla de Determinación de Cloro

Tabla: tbl_parametros

Descripción: En esta tabla registra los campos de la tabla parámetros.

| Campo | Tipo | Nulo | Predeterminado | Descripción | P.Key | F.key |
|---------------------------|-----------|------|----------------|--------------------------------------------|-------|-------|
| regnum | bigserial | No | Null | Generación automática de un nuevo registro | | |
| tbl_tipo_parametro | bigserial | No | Null | Almacena el tipo del parámetro | x | |
| tbl_parametros constantes | bigserial | No | Null | Almacena la constante del parámetro | | x |
| observaciones | text | No | | Almacena las observaciones | | |

Fuente: Propia

Tabla A.4: Tabla de Parámetros

Tabla: tbl_parametros_constantes

Descripción: Contiene los valores de los parámetros constantes ya incluidos en la tabla parámetros.

| Campo | Tipo | Nulo | Predeterminado | Descripción | P.Key | F.key |
|---------------------------|-------------------|------|----------------|-------------------------------------------------------------|-------|-------|
| regnum | bigserial | No | Null | Generación automática de un nuevo registro | x | |
| tbl_constantes | bigserial | No | Null | Almacena las constantes | | x |
| nombre_constante | character varying | No | 100 | Almacena el nombre de la constante | | |
| unidad_medida | character varying | No | 60 | Almacena la unidad de medida | | |
| limite_maximo_permissible | character varying | No | 80 | Almacena el límite máximo permitido según la medida de agua | | |

Fuente: Propia

Tabla A.5: Tabla de Parámetros Contantes

Tabla: tbl_persona

Descripción: Almacena los datos de los usuarios independientemente que sea administrador o usuario.

| Campo | Tipo | Nulo | Predeterminado | Descripción | P.Key | F.key |
|-----------------|-------------------|------|----------------|-------------------------------------------------------------|-------|-------|
| regnum | bigserial | No | Null | Generación automática de un nuevo registro | x | |
| ci_usuario | character varying | No | 11 | Almacena la cedula de identidad del usuario o persona | | |
| nombre_usuario | character varying | No | 100 | Almacena el nombre del usuario o persona | | |
| funcion_usuario | character varying | No | 100 | Almacena la función del usuario sea administrador o usuario | | |

Fuente: Propia

Tabla A.6: Tabla de Persona

Tabla: tbl_registros

Descripción: Almacena todos los registros en la base de datos.

| Campo | Tipo | Nulo | Predeterminado | Descripción | P.Key | F.key |
|-----------------|-------------------|------|----------------|--------------------------------------------|-------|-------|
| regnum | bigserial | No | Null | Generación automática de un nuevo registro | x | |
| tbl_usuarios | bigserial | No | Null | Código del Usuario | | x |
| nombre_tabla | character varying | No | 80 | Nombre de la tabla que modifica | | |
| recnum_registro | bigserial | No | Null | Código del registro | | |
| fecha | character varying | | 25 | Almacena la fecha del registro | | |

Fuente: Propia

Tabla A.7: Tabla de Registros

Tabla: tbl_sector

Descripción: Almacena los sectores que se encuentran distribuidos en toda la ciudad de Ibarra con todos sus datos.

| Campo | Tipo | Nulo | Predeterminado | Descripción | P.Key | F.key |
|---------------|-------------------|------|----------------|--------------------------------------------|-------|-------|
| regnum | bigserial | No | Null | Generación automática de un nuevo registro | x | |
| nombre_sector | character varying | No | 80 | Nombre del Sector | | |
| descripcion | Text | Si | | Descripción de la ubicación del sector | | |

Fuente: Propia

Tabla A.8: Tabla de Sectores

Tabla: tbl_tipo_parametro

Descripción: En esta tabla guardamos los diferentes tipos de parámetros.

| Campo | Tipo | Nulo | Predeterminado | Descripción | P.Key | F.key |
|-----------------------|-------------------|------|----------------|--------------------------------------------|-------|-------|
| regnum | bigserial | No | Null | Generación automática de un nuevo registro | x | |
| nombre_parámetro | character varying | No | 100 | Nombre del Parámetro | | |
| descripcion_parámetro | Text | Si | | Descripción del parámetro | | |

Fuente: Propia

Tabla A.9: Tabla de Tipo de Parámetros

Tabla: tbl_unidades

Descripción: Esta tabla nos muestra las unidades de medida que dispone el Laboratorio.

| Campo | Tipo | Nulo | Predeterminado | Descripción | P.Key | F.key |
|---------------|-------------------|------|----------------|--------------------------------------------|-------|-------|
| regnum | bigserial | No | Null | Generación automática de un nuevo registro | x | |
| nombre_unidad | character varying | No | 100 | Nombre de la unidad | | |
| descripcion | Text | Si | | Descripción de la unidad | | |

Fuente: Propia

Tabla A.10: Tabla de Tipo de Unidades

Tabla: tbl_usuarios

Descripción: Esta tabla nos muestra las unidades de medida que dispone el Laboratorio.

| Campo | Tipo | Nulo | Predeterminado | Descripción | P.Key | F.key |
|-------------|-------------------|------|----------------|--------------------------------------------------------|-------|-------|
| regnum | bigserial | No | Null | Generación automática de un nuevo registro | x | |
| tbl_persona | bigserial | No | Null | Nombre de la Persona | | |
| login | character varying | Si | 100 | El nombre del usuario o el login con que se identifica | | |
| password | character varying | Si | 100 | La contraseña del usuario | | |

Fuente: Propia

Tabla A.11: Tabla Usuarios

9.3 ANEXO C: MANUAL DE INSTALACIÓN

Instalación de herramientas sobre Windows

Requisitos mínimos de hardware:

- Procesador de más de 480 Mhz de velocidad.
- Por lo menos 1Gb de RAM.
- El suficiente espacio de disco duro (por lo menos 5Gb).

PostgreSQL 8.2

Para proceder a la instalación de postgresQL se debe contar con un descompresor, además de esto se debe cambiar las variables de entorno de Windows.

Instalación:

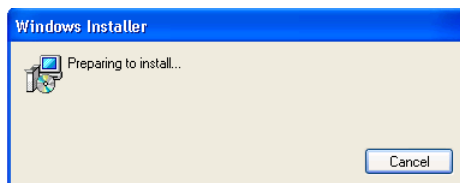
- Arrancamos la instalación a partir del archivo descomprimido.



Fuente: Inatalador de PostgreSQL

Figura 50: Ingreso al Paquete de Instalación

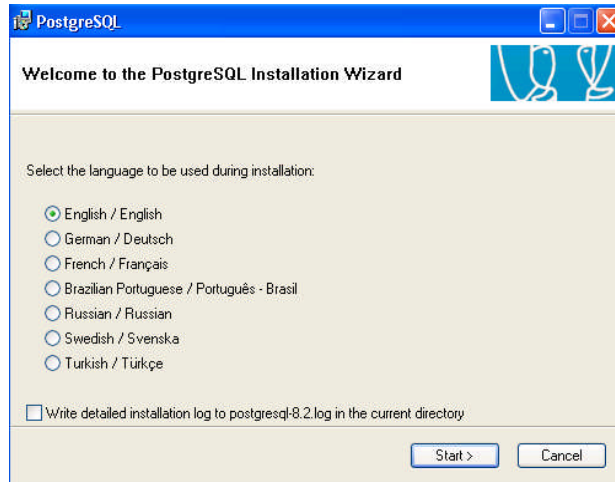
- Esperemos mientras se inicia el instalador



Fuente: Inatalador de PostgreSQL

Figura 51: Mensaje de Preparación de la Instalación

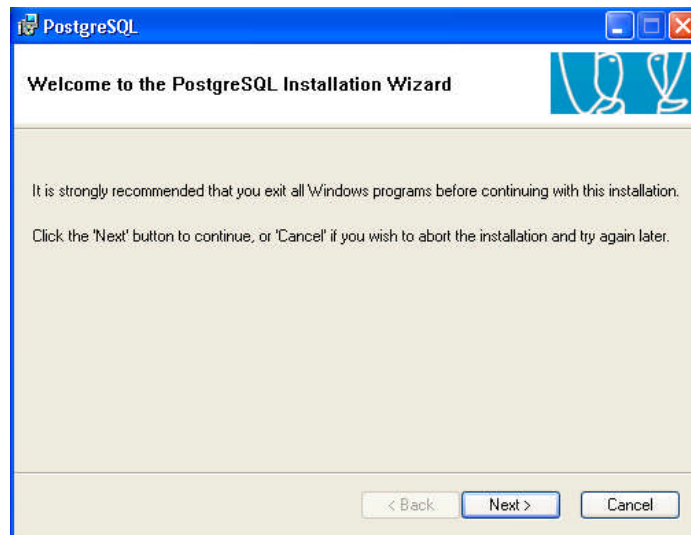
- Seleccionar el idioma para llevar a cabo la instalación



Fuente: Inсталador de PostgreSQL

Figura 52: Selección de Idioma de Instalación

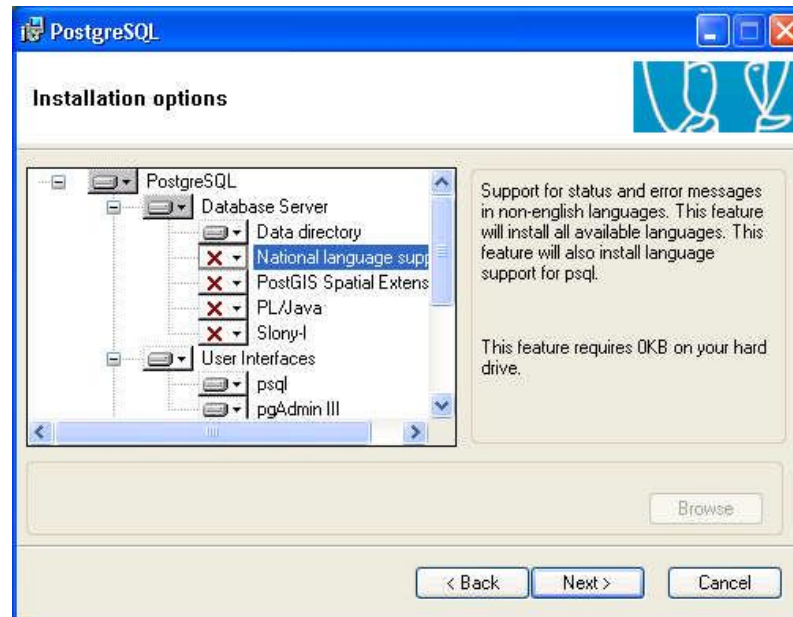
- Pantalla de bienvenida a la instalación de PostgreSQL



Fuente: Inсталador de PostgreSQL

Figura 53: Bienvenida a la Instalación de Programa

- Opciones de paquetes de instalación



Fuente: Instalador de PostgreSQL

Figura 54: Opciones de Instalación

- Inicialización del clúster de la BDD

Se debe configurar lo siguiente:

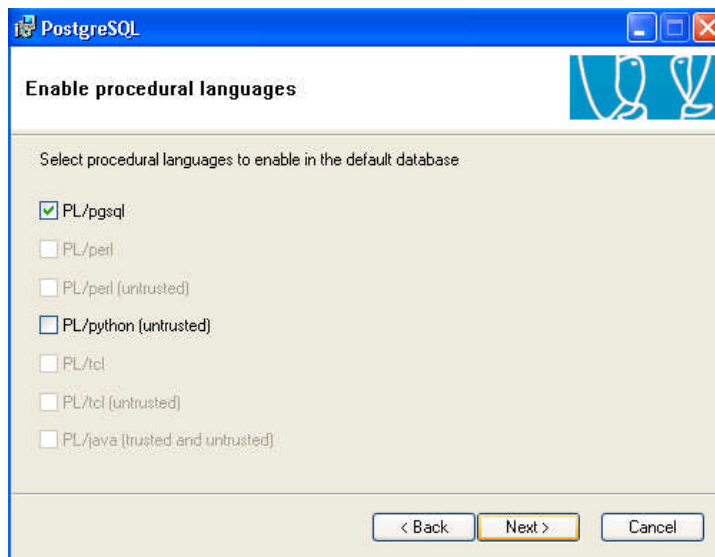
- Puerto.
- Localidad.
- El encoding de la BDD.
- Nombre de superusuario.
- Password.



Fuente: Inatalador de PostgreSQL

Figura 55: Inicialización del Cluster

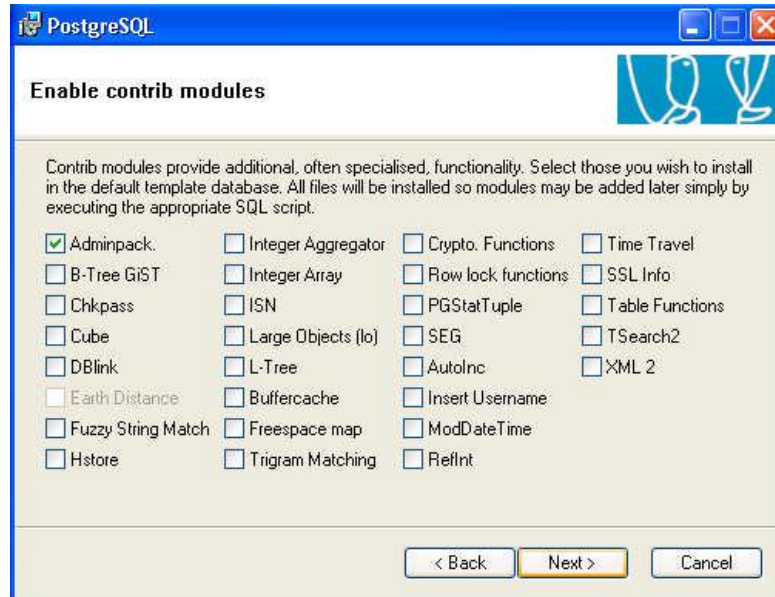
- **Habilitación del lenguaje de procedimiento**



Fuente: Inatalador de PostgreSQL

Figura 56: Habilitar el Lenguaje de Procedimiento

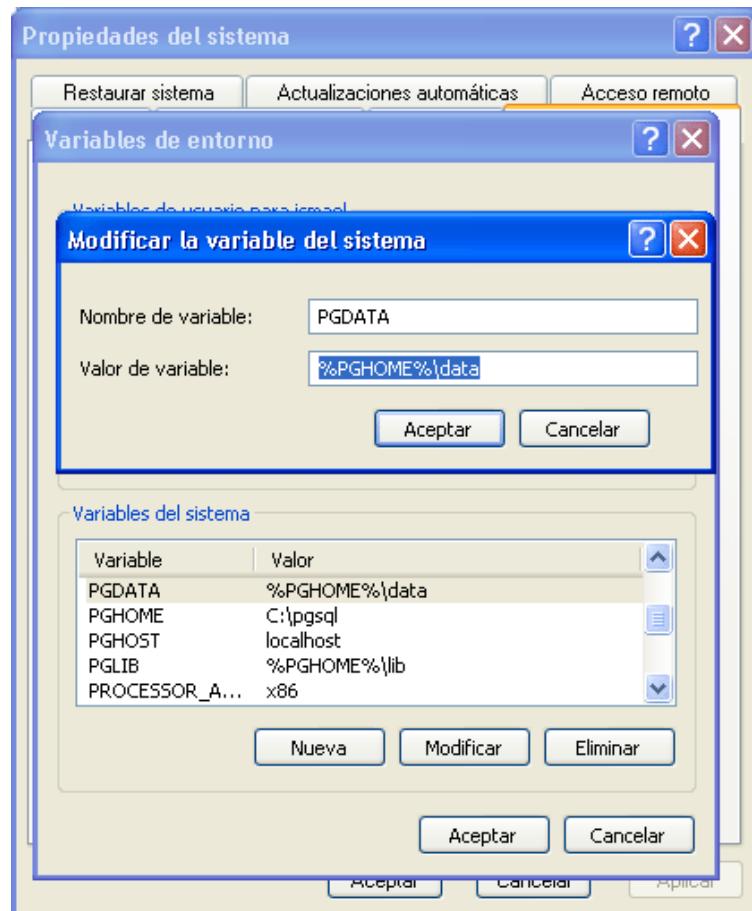
- **Habilitación de módulos contrib**



Fuente: Instalador de PostgreSQL

Figura 57: Habilitación de Módulos Contrib

- Finalizamos la instalación y se debe crear las variables de entorno.



Fuente: Instalador de PostgreSQL

Figura 58: Proceso de Finalización

9.4 ANEXO D: MANUAL DE USUARIO

Conociendo la importancia que tiene para la EMAPA-I se adoptó la creación de un nuevo programa para la ayuda en la administración del laboratorio de Caranqui con el que cuenta y así optimizar recursos tanto económicos como en el mejor uso de los mismos de una manera más fácil.

El manual del usuario del sistema de evaluación, control y reportes del Laboratorio de Caranqui está pensado para ser una guía práctica pero, al mismo tiempo, debe ser un manual de consulta para ayudar al trabajo de los usuarios que deben administrar el Laboratorio de Caranqui.

Explicación de este manual

En este manual se va a encontrar las razones por las que se creó el sistema evaluación, control y reportes del Laboratorio de Caranqui, las ventajas que el mismo aporta a la gestión diaria de la empresa y su funcionamiento y estructura paso a paso.

Empezare por la implantación del propio sistema, explicando sus requisitos y la estructura de sus menús. Procederé después a la exposición del funcionamiento general de las pantallas. Los siguientes bloques nos describirán detalladamente cada uno de los apartados que componen el sistema, su funcionamiento y sus relaciones.

Ventajas del sistema

El sistema evaluación, control y reportes del Laboratorio de Caranqui está estructurado de tal modo que permite al usuario trabajar de una forma simple y con el grado de información que en cada momento necesita.

Los usuarios pueden trabajar y obtener información, pues el sistema de evaluación, control y reportes del Laboratorio de Caranqui devuelve información, evidentemente la información obtenida ira en función del nivel de detalle con que estemos trabajando, el menú que presenta el sistema, son las herramientas esenciales que nos permiten esta flexibilidad en la gestión.

Características del programa

El sistema de evaluación, control y reportes del Laboratorio de Caranqui ha sido diseñado para resolver, de forma práctica y sencilla, la administración del Laboratorio de Caranqui, ofreciendo al usuario todo tipo de ayuda durante su funcionamiento.

Antes de empezar, debemos recordarles los requisitos mínimos del sistema:

- Ordenador Pentium 1Ghz o superior
- 128 Mb de RAM
- 50 Mb libres en disco duro
- Conexión a Internet

Antes de proceder a la exposición del método de implantación del aplicativo, resaltando sus características más esenciales, como su fácil manejo, la aplicación modular, su flexibilidad y adaptabilidad y su apertura e interconexión. El sistema de evaluación, control y reportes del Laboratorio de Caranqui es fácil de manejar gracias a sus menús sencillos y bien definidos.

La adaptabilidad del sistema de evaluación, control y reportes del Laboratorio de Caranqui se basa en que toda la información se gestiona en un servidor el cual debe prestar las seguridades pertinentes, el servidor debe disponer de un ancho de banda de más de 10Mbytes, para que garantice una comunicación rápida y fluida hacia la aplicación en la máquina cliente.

Estructura del programa

El sistema de monitoreo y localización vehicular, está compuesto primeramente por una pantalla donde los usuarios deben logearse para poder tener acceso al mismo.



Fuente: Propia del Sistema

Figura 59: Estructura de Inicialización del Sistema

Una vez registrado el login y el password ya sea como administrador o usuario ingresa a las pantallas según su grado de permisos que le haya otorgado el sistema realizando un clic en el botón Ingresar.

- Realizado el ingreso como administrador se tiene los privilegios del manejo entero del sistema como son todas las pantallas que nos muestra a continuación:



Fuente: Propia del Sistema

Figura 60: Menú Principal de Sistema

- A continuación nos muestra el funcionamiento de cada uno de los accesos del sistema :

Ingreso Tipo Variables

Realizando clic en este enlace nos aparece la pantalla para realizar el ingreso de tipos de Variables y realizar sus respectivas modificaciones como nos indica a continuación.

The image shows a web interface for variable management. It consists of two stacked panels. The top panel is titled 'ADMINISTRACION DEL CATALOGO DE VARIABLES' and contains a sub-section 'Ingreso de Datos' with a button labeled 'Ingresar'. The bottom panel is titled 'MODIFICACION DEL CATALOGO DE VARIABLES' and contains a sub-section 'Actualizar Variable'. This section includes a dropdown menu for 'Nombre de la Variable' with 'Radiactivos' selected, and a button labeled 'Modificacar Variable'.

Fuente: Propia del Sistema

Figura 61: Administración del Catalogo de Variables

Ingresar

Para poder realizar un ingreso nuevo de variables realizamos un clic en el botón Ingresar y nos aparece la pantalla siguiente:

| INGRESAR DATOS DE LAS VARIABLES | |
|--------------------------------------------------------------------------------------|----------------------|
| Nombre de la Variable : | <input type="text"/> |
| Descripcion de la Variable: | <input type="text"/> |
| <input type="button" value="Ingresar"/> Principal Catalogo Variables | |

Fuente: Propia de Sistemas

Figura 62: Tabla para Ingreso de Datos de Variables

Una vez realizado el ingreso de los datos realizamos un clic y los datos se almacenan automáticamente en la base de datos.

[Principal Catalogo Variables](#)

Si queremos regresar al menú de catalogo de variables hacemos un clic en el link.

Para poder realizar la modificación a los datos ya existentes hacemos un clic en modificar variable en la pantalla de ingreso de tipo de variable.

| DATOS DE LAS VARIABLES | |
|-------------------------------------------------------------------------------------------|----------------------|
| ID de la Variable | 2 |
| Nombre de la Variable : | Radiactivos |
| Descripcion de la Variable | <input type="text"/> |
| <input type="button" value="Actualizar"/> <input type="button" value="Regresar al Menu"/> | |

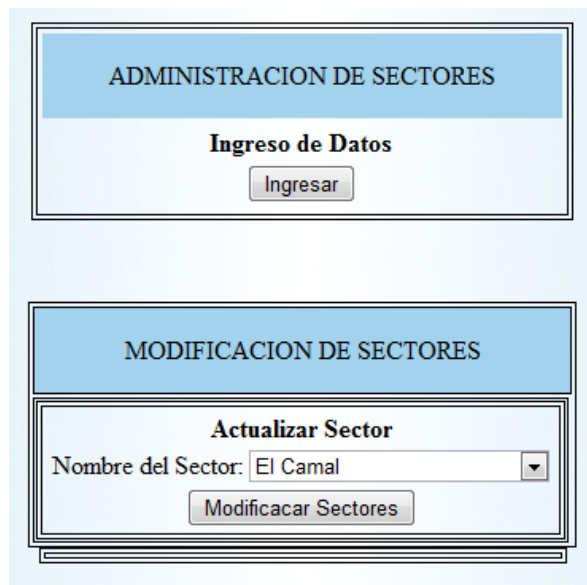
Fuente: Propia de Sistemas

Figura 63: Ingreso de Datos de Las Variables

Actualizar Al realizar un clic en este botón los datos se almacenan directamente en la base de datos.

Regresar al Menu Regresa a la pantalla de principal catalogo variable.

Ingreso Sectores Dando un clic en este botón nos muestra las pantallas de administración e sectores y modificación de sectores.



Fuente: Propia del Sistema

Figura 64: Administración del Sectores

Ingresar Realizando un clic en el botón ingresar nos muestra la página de ingreso de los sectores.

| INGRESAR DATOS DE LOS SECTORES | |
|----------------------------------------------------------------------------|----------------------|
| Nombre del Sector : | <input type="text"/> |
| Descripcion del Sector: | <input type="text"/> |
| <input type="button" value="Ingresar"/> Principal Sectores | |

Fuente: Propia del Sistema

Figura 65: Ingreso de Datos de los Sectores

Al hacer clic en este botón nos guarda los datos directamente en la base de datos.

[Principal Sectores](#) Al realizar clic en este link nos lleva a la página principal de ingreso de Sectores.

Al hacer clic en este link nos lleva a la siguiente página donde nos muestra los datos que tenemos que modificar.

| DATOS DEL SECTOR | |
|-------------------------------------------------------------------------------------------|--------------------------------------|
| ID del Sector: | 9 |
| Nombre del Sector : | El Camal |
| Descripcion del Sector: | Lugar para determinar cloro residual |
| <input type="button" value="Actualizar"/> <input type="button" value="Regresar al Menu"/> | |

Fuente: Propia del Sistema

Figura 66: Actualización de Datos de Sectores.

Actualizar

Al hacer un clic en este botón nos actualiza los datos en la base de datos.

Regresar al Menu

Este botón los regresa a la pantalla de ingreso de sectores.

[Parametros
Constantes](#)

En este link nos muestra la pantalla principal de los parámetros constantes donde se encuentra la administración de parámetros y la modificación del los parámetros constantes.

ADMINISTRACION DE PARAMETROS

Ingresar Nuevos Parametros a las Variables

Ingresar

MODIFICACION DE PARAMETROS DE LAS VARIABLES

Actualizar Parametros de Variables

Nombre de la Variables:

Dureza Total, CaCO3

Modificacar Parametros de Variables

Fuente: Propia del Sistema

Figura 67: Administración de Parámetros.

Ingresar

Quando hacemos un clic en el botón ingresar nos despliega la siguiente pantalla en donde se realiza el ingreso de los parámetros de las variables.

| INGRESAR PARAMETROS DE LAS VARIABLES | |
|------------------------------------------------------------------------------|-------------|
| Tipo de Variable | Radiactivos |
| Nombre de la constante : | |
| Unidad de Medida : | |
| Limite Maximo Permissible : | |
| <input type="button" value="Ingresar"/> Principal parametros | |

Fuente: Propia del Sistema

Figura 68: Ingreso de Parámetros de Variables

Una vez llenado los campos de esta tabla realizamos un clic en este botón e ingresa los datos en la base de datos.

[Principal parametros](#)

Si queremos regresar a la pantalla principal de parámetros realizamos un clic en este botón.

Al ingresar a la modificación de parámetros de variables obtenemos la siguiente pantalla donde nos indica los datos que podemos modificar.

| DATOS DE LOS PARAMETROS DE LAS VARIABLES | |
|-------------------------------------------------------------------------------------------|---------------------|
| ID del Parametro de la Variable: | 15 |
| Tipo de Variable : | Inorganicos |
| Nombre de la Constante : | Dureza Total, CaCO3 |
| Unidad de Medida : | mg/l |
| Limite Maximo Permissible : | 300 |
| <input type="button" value="Actualizar"/> <input type="button" value="Regresar al Menu"/> | |

Fuente: Propia del Sistema

Figura 69: Datos de los Parámetros de las Variables

Actualizar

Realizando un clic en este botón nos actualiza de manera automática los datos modificados en la base de datos.

Regresar al Menu

Al hacer clic en este botón nos regresa al menú de parámetros de variables.

**Ingreso Datos
Determinacion de
Cloro Residual**

Al hacer clic en este link nos despliega una pantalla para poder realizar el ingreso de datos y observar las tablas de datos ingresados.



Fuente: Propia del Sistema

Figura 70: Administrador de Cloro Residual.

Ingresar

Al realizar un clic en este botón nos despliega una pantalla para tener acceso al ingreso de nuevos datos.

| DETERMINACION DE CLORO RESIDUAL EN DIFERENTES PUNTOS | |
|------------------------------------------------------------------------------------------------------|------------|
| Nombre del Sector: | El Camal |
| Fecha (MM-dd-yyyy) : | 16/07/2011 |
| Hora (h:m): | |
| Cantidad de Cloro : | 0.0 |
| <input type="button" value="Ingresar"/> Principal Tabla determinacion Cloro Residual | |

Fuente: Propia del Sistema

Figura 71: Determinación de Cloro Residual - Ingreso

Si ya esta lista la información en todos los campos y deseamos que se guarde en la base de datos realizamos un clic en este botón.

[Principal Tabla determinacion Cloro Residual](#)

Si queremos regresar a la pantalla principal determinación de cloro residual realizamos un clic en este botón.

Si queremos visualizar los datos en la tabla de cloro residual hacemos un clic en este botón y nos despliega la siguiente pantalla.

| DATOS PARA OBTENER LA DETERMINACION DEL CLORO RESIDUAL | |
|-----------------------------------------------------------------------------------------|------------|
| Fecha de Inicio : | 2008-06-16 |
| Fecha de Fin : | 2008-06-16 |
| Nombre del Sector : | El Camal |
| <input type="button" value="Reporte"/> Principal Determinacion de Cloro | |

Fuente: Propia del Sistema

Figura 72: Ingreso Reportes de Cloro Residual

Reporte

Primeramente para obtener un reporte debemos de seleccionar las fechas de inicio, fecha de fin y el nombre del sector.

Principal Determinacion de Cloro

En este link nos regresa a la página principal de cloro residual.

Ingreso Tipo de Parametros

Cuan do realizamos un clic en este link nos lleva a una página donde nos muestra la administración del tipo de parámetros y sus respectiva modificación.

The image shows a screenshot of a web application interface. It is divided into two main sections. The top section is titled 'ADMINISTRACION DE TIPO DE PARAMETROS' and contains a sub-section 'Ingreso de Tipo de Parametros' with an 'Ingresar' button. The bottom section is titled 'MODIFICACION DE TIPO DE PARAMETROS' and contains a sub-section 'Actualizar Tipo de Parametro'. This section includes a label 'Nombre del Tipo de Parametro' followed by a dropdown menu currently displaying 'Analisis Fisicos' and a 'Modificacar Variable' button.

Fuente: Propia del Sistema

Figura 73: Administración de Tipos de Parámetros.

Ingresar

Al hacer clic en este botón nos muestra una pantalla donde tenemos que realizar el ingreso de nuevos datos.

INGRESAR DATOS DE LOS TIPOS DE PARAMETROS

Nombre del Tipo de Parametro :

Descripcion del Tipo de Parametro:

Ingresar [Principal Tipo Parametros](#)

Fuente: Propia del Sistema

Figura 74: Ingreso de Datos de Tipos de Parámetros

Ingresar

Una vez llenado los campos que observamos en esta tabla vamos a guardarlos en la base de datos haciendo un clic en el botón ingresar.

[Principal Tipo Parametros](#)

Si queremos regresarnos al menú de tipo de parámetros hacemos un clic en este botón.

Modificacar Variable

Quando vamos a realizar una modificación a nuestros datos de la tabla tipos de parámetros hacemos un clic en este botón y nos muestra la siguiente tabla.

DATOS DE LOS TIPOS DE PARAMETROS

ID del Tipo de Parametro 1

Nombre del Tipo de Parametro : Analisis Fisicos

Descripcion del Tipo de Parametro

Actualizar Regresar al Menu

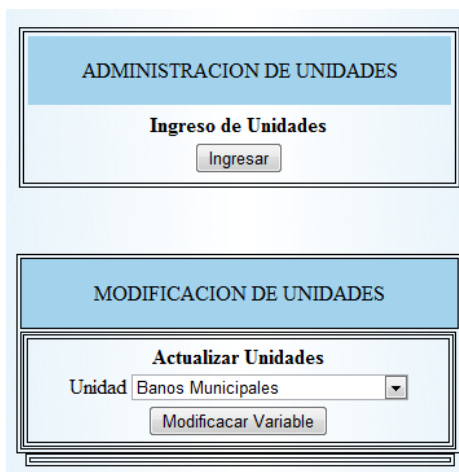
Fuente: Propia del Sistema

Figura 75: Actualización de Tipos de Parámetros.

Actualizar Una realizada la modificación en los datos procedemos a guardar en la base de datos realizando un clic en el botón actualizar.

Regresar al Menu Este botón nos lleva a la página principal de tipo de parámetros.

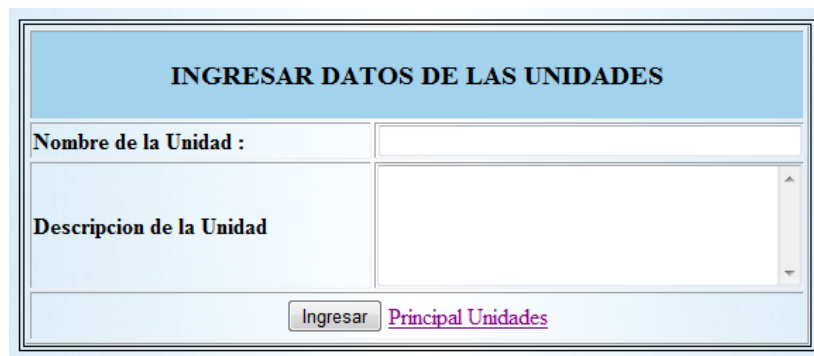
Ingreso de Unidades Este enlace nos lleva a la paina de unidades que muestra la siguiente pantalla.



Fuente: Propia del Sistema

Figura 76: Administración de Unidades

Ingresar Si realizamos un clic en el botón ingresar nos lleva a la página siguiente.



Fuente: Propia del Sistema

Figura 77: Ingreso de Datos de las Unidades

Ingresar Una vez llenado todos los datos en los campos procedemos a guardar en la base de datos haciendo un clic en el botón ingresar.

Principal Unidades Si queremos regresar al menú principal de unidades hacemos clic en este link.

Modificacar Variable Si tenemos llenado valores en la tabla de unidades y queremos modificarlos realizamos un clic en este botón y nos presenta la siguiente tabla.

| DATOS DE LAS UNIDADES | |
|-------------------------------------------------------------------------------------------|-------------------|
| ID de la Unidad | 3 |
| Nombre de la Unidad : | Banos Municipales |
| Descripcion de la Unidad | Guaracazapas |
| <input type="button" value="Actualizar"/> <input type="button" value="Regresar al Menu"/> | |

Fuente: Propia del Sistema

Figura 78: Actualización de Unidades.

Actualizar Una vez realizadas las debidas correcciones procedemos a guardar los datos en la base de datos haciendo un clic en el botón actualizar.

Regresar al Menu Si queremos regresar al menú de unidades hacemos clic en este botón.

Ingreso de Parametros Realizando un clic en este link observamos la siguiente pantalla que nos muestra el ingreso de datos nuevos y sus respectivas tablas.

ADMINISTRACION DE PARAMETROS

Ingreso de Parametros

Ingresar

MODIFICACION DE SECTORES

Visualizar Informacion

Nombre de Parametros: Analisis Fisicos

Ver Datos

Fuente: Propia del Sistema

Figura 79: Modificación de Sectores.



Al realizar un clic en este botón nos ingresa a la siguiente pantalla.

INGRESAR PARAMETROS

Tipo de Parametro: Analisis Fisicos

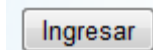
Tipo de Constante: Dureza Total, CaCO3

Observacion :

Ingresar [Principal parametros](#)

Fuente: Propia del Sistema

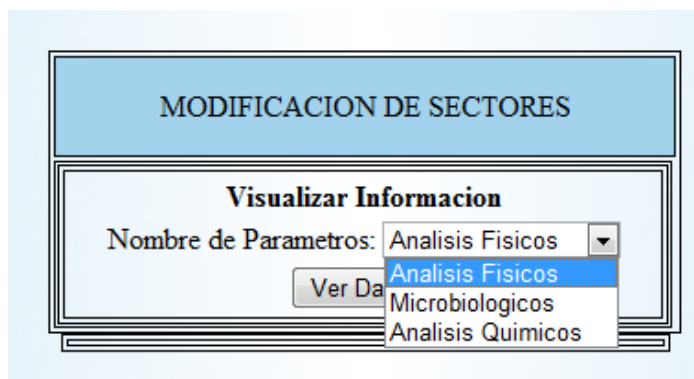
Figura 80 :Ingreso de Parametros



Este botón nos permite registrar los valores nuevos en la base de datos.

Principal parametros

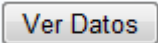
Este botón nos permite regresarnos a la pantalla principal de ingreso de parámetros.




Fuente: Propia del Sistema

Figura 81 :Modificasiónn de Sectores

Una vez seleccionada la opción de parámetro que queremos visualizar los valores de las tablas realizamos un clic en el botón Ver Datos

 Cuando realizamos un clic en este botón nos despliega la siguiente pantalla donde observamos todos los datos referentes al valor seleccionado.

| Tipo de Parametro Analisis Fisicos | |
|------------------------------------|--------------------------------------------------------------|
| Nombre de Variable | Observaciones |
| Color | apariencia fisica |
| Turbiedad | afecta la apariencia fisica interfiere en la desinfeccion |
| pH | 7 y 8.5 agua alcalina |
| Alcalinidad Total | |



Fuente: Propia del Sistema

Figura 82: Tabla de Parámetros.

Regresar al Menu

Este botón nos permite regresarnos al menú principal de parámetros.

Ingreso de Analisis

Este link despliega la pantalla en donde se encuentra el ingreso de nuevos valores de análisis y sus respectivas tablas.

INGRESO DE DATOS PARA ANALISIS DE TOMA DOMICILIARIA

Ingreso de Datos

Ingresar

TABLA DE VALORES INGRESADOS

Ver Tabla de Valores

Visualizar

Fuente: Propia del Sistema

Figura 83: Ingreso de Datos de Tomas Domiciliaras

Ingresar

Realizando un clic este botón nos despliega la tabla siguiente.

INGRESAR PARAMETROS

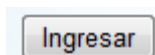
| | |
|------------------------|---------------------|
| Nombre de la Constante | Dureza Total, CaCO3 |
| Unidad | Banos Municipales |
| Fecha : | 17/07/11 |
| Ingresar Valor : | 0 |

Ingresar

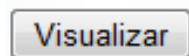
Fuente: Propia del Sistema

Figura 84: Ingreso de Parámetros

En esta tabla tiene campos de selección en donde una vez realizada la selección correcta procedemos a ingresar los demás valores.



Cuando hacemos un clic en este botón nos guarda todos los datos de la tabla en la base de datos.



Si queremos visualizar los datos de las diferentes tablas nos despliega la siguiente pantalla.

DATOS PARA OBTENER EL ANALISIS A NIVEL DE TOMA DOMICILIARIA

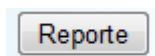
| | | |
|-------------------|------------|---|
| Fecha de Inicio : | 2009-02-02 | ▼ |
| Fecha de Fin : | 2009-02-02 | ▼ |

Reporte [Principal Determinacion de Analisis](#)

Fuente: Propia del Sistema

Figura 85: Reportes de Datos de Tomas Domiciliaras

Para obtener un reporte primeramente seleccionamos las fechas de inicio y fin.



Este botón nos permite obtener reportes en forma visual.

[Principal Determinacion de Analisis](#)

Este botón nos permite regresar a la pantalla principal de determinación de análisis.

| ANÁLISIS A NIVEL DE TOMA DOMICILIARIA: SISTEMA CARANQUI | | | | | |
|----------------------------------------------------------------------------------------|----------|-------------|----|------------|-------------------|
| Parametros | Unidades | U. Catolica | | 2009-02-02 | |
| Análisis Físicos | | | | | |
| 1 | Color | UTC | 15 | 0 | apariencia fisica |
| Microbiologicos | | | | | |
| Análisis Quimicos | | | | | |
| <input type="button" value="Grafica"/> <input type="button" value="Regresar al Menu"/> | | | | | |

Fuente: Propia del Sistema

Figura 86: Tabla de Análisis de Tomas Domiciliaras

Este botón nos permite observar los datos en forma grafica.

Este botón nos permite regresar a la pantalla principal de ingresos de análisis.

