

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**DESARROLLO DE UNA PLATAFORMA WEB PARA EL SISTEMA
DE GESTIÓN DE LA INFORMACIÓN DE PROYECTOS DE
FISCALIZACIÓN REALIZADOS POR LA EMPRESA TECNIE,
ACCESIBLE LOCAL Y REMOTAMENTE**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y REDES DE INFORMACIÓN**

SIGIFREDO PATRICIO CAMACHO CASTILLO
pcamacho7@gmail.com

DIRECTOR: Ing. Williams Fernando Flores Cifuentes
fernando.flores@epn.edu.ec

Quito, Junio 2015

DECLARACIÓN

Yo Sigifredo Patricio Camacho Castillo, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Sigifredo Patricio Camacho Castillo

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Sigifredo Patricio Camacho Castillo, bajo mi supervisión.

Ing. Fernando Flores
DIRECTOR DEL PROYECTO

AGRADECIMIENTOS

A mi padre Sigifredo Camacho, quien siempre me apoyó y lo sigue haciendo a lo largo de mi carrera y mi desarrollo personal en todas las formas posibles.

A mi hermano Diego Camacho por darme una mano cuando necesite de su ayuda.

Patricio

DEDICATORIA

A mi padre gracias por estar siempre ya sea lejos o cerca.

Patricio

CONTENIDO

DECLARACIÓN	I
CERTIFICACIÓN	II
AGRADECIMIENTOS	III
DEDICATORIA.....	IV
CONTENIDO.....	V
ÍNDICE DE TABLAS	XII
ÍNDICE DE FIGURAS	XIV
RESUMEN	XVII
PRESENTACIÓN	XVIII
CAPÍTULO 1	1
FUNDAMENTOS TEÓRICOS	1
1.1 METODOLOGÍAS DE DESARROLLO WEB	1
1.1.1 METODOLOGÍAS DE DESARROLLO TRADICIONAL.....	1
1.1.2 METODOLOGÍAS DE DESARROLLO ÁGIL.....	1
1.1.3 COMPARACIÓN DE METODOLOGÍAS TRADICIONALES Y ÁGILES .	2
1.1.4 CARACTERÍSTICAS FUNDAMENTALES DE LA METODOLOGÍA XP	2
1.1.4.1 Simplicidad de código.....	2
1.1.4.2 Refactorización del código.....	3
1.1.4.3 Integración frecuente de desarrollador-usuario	3
1.1.4.4 Corrección total de errores	3
1.1.4.5 Pruebas unitarias continuas	3
1.1.4.6 Roles XP	3
1.1.4.6.1 Programador	3
1.1.4.6.2 Jefe de proyecto.....	3
1.1.4.6.3 Cliente	4
1.1.4.7 Ciclo de vida de la metodología XP	4
1.1.4.7.1 Etapa de exploración	4
1.1.4.7.2 Etapa de planificación de entrega	4
1.1.4.7.3 Etapa de Iteraciones	4
1.1.4.7.4 Etapa de producción	5
1.1.4.7.5 Finalización del sistema	5
1.2 PLATAFORMAS Y SERVIDORES WEB	5
1.2.1 PLATAFORMA WEB	5
1.2.1.1 Definición de plataforma Web.....	5

1.2.1.2 Plataforma	5
1.2.1.3 Plataforma de desarrollo de software	5
1.2.1.3.1 LAMP	6
1.2.1.3.2 WAMP	7
1.2.1.3.3 XAMP	7
1.2.1.3.4 WXCF.....	7
1.2.1.3.5 WISA.....	8
1.2.2 SERVIDORES WEB.....	9
1.2.2.1 Internet	9
1.2.2.2 El servicio Web	9
1.2.2.2.1 Hipertexto.....	9
1.2.2.3 Relación cliente/servidor.....	10
1.2.2.3.1 Servidores	10
1.2.2.3.2 Clientes	10
1.2.2.4 Servidor Web	10
1.2.2.4.1 Programa	10
1.3 GESTORES DE CONTENIDO	11
1.4 LENGUAJE DE PROGRAMACIÓN	12
1.4.1 TIPOS DE LENGUAJES DE PROGRAMACIÓN DE ALTO NIVEL	12
1.4.1.1 Lenguajes compilados	12
1.4.1.2 Lenguajes interpretados	12
1.5 DESARROLLO DEL LADO DEL SERVIDOR Y CLIENTE EN AMBIENTES WEB	12
1.5.1 DESARROLLO DEL LADO DEL SERVIDOR	13
1.5.2 DESARROLLO DEL LADO DEL CLIENTE	13
1.5.3 DESARROLLO DEL LADO DEL CLIENTE VS SERVIDOR.....	13
1.5.3.1 Lenguajes de programación del lado del servidor	15
1.5.3.1.1 PHP	15
1.5.3.1.2 ASP: Páginas de Servidor Activas	15
1.5.3.1.3 JSP: Páginas de Servidor Java	16
1.5.3.2 Lenguajes de programación del lado del cliente.....	16
1.5.3.2.1 Java	16
1.5.3.2.2 VBScript	17
1.5.3.2.3 Javascript.....	17
1.6 BASE DE DATOS RELACIONALES.....	18
1.6.1 BASE DE DATOS	18
1.6.1.1 Base de datos relacionales.....	18

1.7 SISTEMAS DE GESTIÓN DE BASES DE DATOS	19
1.7.1 MYSQL.....	19
1.7.2 SQL Server	20
1.7.3 ORACLE	20
CAPÍTULO 2	21
DISEÑO E IMPLEMENTACIÓN: PLATAFORMA WEB.....	21
2.1 ESTABLECIMIENTO DE LÍNEA BASE	21
2.1.1 HISTORIA	21
2.1.2 ANTECEDENTES DEL PROBLEMA	21
2.1.3 SOLUCIÓN PROPUESTA	22
2.2 ESTUDIO PRELIMINAR DEL MANEJO, CONTROL Y GESTIÓN DE INFORMACIÓN/DOCUMENTACIÓN DE LA EMPRESA TECNIE	22
2.2.1 ANTECEDENTES	22
2.2.1.1 Proceso de desarrollo de un proyecto en TECNIE:	23
2.2.1.1.1 Etapas del proyecto:	23
2.2.1.2 Sistema de control de documentación.....	29
2.2.1.2.1 Tipo de documentación manejada	31
2.2.1.2.2 Codificación.....	31
2.2.1.2.3 Almacenamiento de documentación/información	31
2.2.1.2.4 Distribución de la información	32
2.2.1.2.5 Identificación de documentos.....	32
2.2.2 SELECCIÓN DE METODOLOGÍA DE DESARROLLO.....	32
2.2.2.1 Planificación del proyecto usando la metodología elegida:	33
2.3 INICIALIZACIÓN DE DESARROLLO DEL PROYECTO	34
2.3.1 EQUIPO DE TRABAJO.....	34
2.3.2 REQUERIMIENTOS FUNCIONALES	35
2.3.2.1 Ingresar/Almacenar/Gestionar:	35
2.3.2.2 Generar/Proporcionar:	35
2.3.3 REQUERIMIENTOS NO FUNCIONALES:.....	35
2.3.4 DIMENSIONAMIENTO DE HARDWARE	36
2.3.4.1 Cantidad de Información estimada por cada proyecto realizado ...	36
2.3.4.2 Cantidad de usuarios que utilizaran el sistema.....	36
2.3.4.2.1 Análisis de requisitos de rendimiento	37
2.3.5 ALCANCE DEL SOFTWARE	38
2.3.6 PERFILES DE USUARIOS DEL SISTEMA.....	39
2.3.6.1 Usuarios internos	39
2.3.6.1.1 Administrador:	39

2.3.6.1.2 Editor:.....	39
2.3.6.1.3 Lector:	40
2.3.7 JUSTIFICACIÓN DE LA SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO.....	40
2.3.7.1 Servidor Web: Apache 2.2.8:.....	41
2.3.7.2 Gestor de base de datos: MySQL 5.0.51b:.....	41
2.3.7.3 Lenguaje de programación - Cliente: JavaScript:	41
2.3.7.4 Lenguaje de programación – Servidor: PHP 5.2.6:.....	41
2.3.7.5 Entorno de desarrollo integrado: Netbeans 8.0.2 y Notepad++ 6.7.3	42
2.3.8 HISTORIAS DE USUARIO.....	42
2.3.8.1 Iteraciones	43
2.3.8.1.1 Primera iteración	43
2.3.8.2 Estimación de tiempo y esfuerzos	45
2.3.8.2.1 Aspectos a considerarse:	45
2.3.8.3 Planificación de entregas.....	48
2.3.8.3.1 Planificación de entregas de historias de usuario	48
2.3.8.3.2 Planificación de entregas de Historias de usuario y Tareas de usuario:	49
2.3.8.3.3 Análisis y tiempos totales de entrega por iteraciones	50
2.4 DISEÑO DEL PROTOTIPO DE SOFTWARE.....	51
2.4.1 ELECCIÓN DE LA ARQUITECTURA	51
2.4.2 DIAGRAMA DE CLASES	52
2.4.3 DIAGRAMA DE BASES DE DATOS	54
2.4.3.1 Diccionario de datos	55
2.4.4 DISEÑO DE INTERFAZ GRÁFICA DEL USUARIO	59
2.4.4.1 Spikes	59
2.4.4.2 Características generales de una interfaz gráfica	60
2.4.4.3 Características principales de la interfaz gráfica de SGI	60
2.4.4.3.1 Características del usuario lector	60
2.4.4.3.2 Características del usuario editor	61
2.4.4.3.3 Características del usuario administrador	61
2.4.4.4 Diseño e implementación de interfaz gráfica usando hojas de estilo	66
2.4.4.4.1 Cabecera.....	66
2.4.4.4.2 Pantalla inicial	67
2.4.4.4.3 Ingreso de credenciales	68

2.4.4.4.4 Menú principal	69
2.4.4.4.5 Vista de consulta	70
2.4.4.4.6 Módulo de Edición.....	72
2.4.4.4.7 Módulo de administrador (edición de usuarios):.....	74
2.4.4.5 Iconos, botones y enlaces	76
2.4.4.6 Mensajes emergentes de aviso	77
2.4.5 CODIFICACIÓN DE SOFTWARE	79
2.4.5.1 Estándares de programación.....	79
2.4.5.1.1 PHP	79
2.4.5.1.2 MySQL	80
2.4.5.1.3 Clases	80
2.4.5.1.4 Funciones y métodos:	81
2.4.5.2 Implementación del código de las clases del sistema	81
2.4.5.2.1 Clase seguridad	81
2.4.5.2.2 Clase conexión.....	83
2.4.5.2.3 Clase registro	84
2.4.5.2.4 Clase ingresodatos	90
2.4.5.3 CSS – Hojas de estilo.....	94
2.4.5.4 JavaScript - Código	94
CAPÍTULO 3.	96
SEGURIDADES, GESTIÓN DE TRANSACCIONES Y RESPALDOS	96
3.1 SEGURIDADES PARA EL SISTEMA.....	96
3.1.1 SEGURIDADES A NIVEL DEL APlicATIVO WEB	97
3.1.1.1 Validación de acceso a módulos y vínculos Web	97
3.1.1.2 Autenticación de credenciales y encriptación de contraseña	97
3.1.1.3 Envío de variables dentro de PHP	98
3.1.1.4 Establecimiento de Timeout de sesión	98
3.1.2 SEGURIDADES A NIVEL DE LA RED.....	98
3.1.2.1 Sugerencias de software de la infraestructura de red.....	99
3.1.2.1.1 Sistema operativo del servidor principal.....	99
3.1.2.1.2 Reglas para firewall software del sistema operativo	100
3.1.2.1.3 Reglas de port forwarding en el sistema operativo	100
3.1.2.1.4 Dispositivos de conexión inalámbrica interna.....	101
3.1.3 RECOMENDACIONES PARA SOFTWARE DEL APlicATIVO	101
3.1.3.1 Apache Server y PHP	101
3.1.3.1.1 Ocultamiento de información del software del aplicativo.....	101

3.1.3.1.2 Bloquear el acceso directo a archivos mediante URL	102
3.1.3.2 MySQL Server y PHP	102
3.1.3.2.1 Cambio de contraseña de del usuario root en MySQL(durante la instalación).....	102
3.1.3.2.2 Creación de un usuario en MySQL con permisos restringidos	102
3.1.3.2.3 Restricción de acceso por IP	103
3.1.4 RECOMENDACIONES PARA ASEGURAMIENTO DE RED FÍSICA	103
3.1.4.1 Delegación y asignación de Recurso humano apto.....	103
3.1.4.2 Administrativo: Definición de procedimientos y Contraseñas robustas	103
3.1.4.3 Seguridad física	104
3.1.4.4 Análisis de vulnerabilidades y plan de contingencia ante fallos...	104
3.1.4.5 Preparación de un plan de contingencia.....	104
3.2 GESTIÓN Y MONITOREO DE TRANSACCIONES	104
3.3 RESPALDOS DE BASES DE DATOS.....	106
3.3.1 CREACIÓN DE UN ARCHIVO BATCH.....	106
3.3.2 PROGRAMACIÓN DE BACKUP DE BASES DE DATOS MANUAL..	107
CAPÍTULO 4	110
PRUEBAS, ANÁLISIS DE RESULTADOS Y COSTOS	110
4.1 PRUEBAS DEL SOFTWARE	110
4.1.1 PRUEBAS DE ACEPTACIÓN	111
4.1.2 EVALUACIÓN DE RESULTADOS	113
4.1.2.1 Velocidad de transacciones	113
4.1.2.2 Concurrencia: Acceso al sistema e ingreso de información	113
4.2 ANÁLISIS DE RESULTADOS Y COSTOS DE LA APLICACIÓN	114
4.2.1 RESULTADOS DE LA APLICACIÓN	114
4.2.1.1 SUS (System Usability Scale).....	114
4.2.1.1.1 Forma de cálculo.....	114
4.2.2 COSTOS DE LA APLICACIÓN	115
4.2.2.1 Costos de hardware	116
4.2.2.2 Costos de Software para desarrollo.....	116
4.2.2.3 Costos de licencia para desarrollo	116
4.2.2.4 Costos de desarrollo	116
4.2.2.5 Costos de aprendizaje	116
4.2.2.6 Cálculo de costos	116
CAPÍTULO 5	118

CONCLUSIONES y RECOMENDACIONES	118
5.1 CONCLUSIONES	118
5.2 RECOMENDACIONES.....	120
REFERENCIAS BIBLIOGRÁFICAS	123
ANEXOS	126

ÍNDICE DE TABLAS

Tabla 1.1: Diferencia entre metodologías tradicionales y ágiles.....	2
Tabla 1.2: Diferencias entre la programación del lado del cliente y del lado del servidor.....	14
Tabla 2.1: Historia de usuario 1: Acceso al sistema.....	44
Tabla 2.2: Tarea de usuario 1: Acceder al sistema por medio login (Validación de datos de usuario).....	44
Tabla 2.3: Historia de usuario 2: Gestión de contenedor: proyecto.....	44
Tabla 2.4: Tarea de usuario 2: Crear nuevo registro.....	44
Tabla 2.5: Tarea de usuario 3: Listar registros existentes.....	44
Tabla 2.6: Tarea de usuario 4: Modificar registro.....	45
Tabla 2.61: Estimación de esfuerzos.....	46
Tabla 2.62: Estimación de esfuerzos con tiempos adicionales.....	47
Tabla 2.63: Historias de usuario con iteraciones.....	48
Tabla 2.64: Planificación de entrega de historias de usuario.....	49
Tabla 2.65: Fechas de planificación de entrega de historias y tareas de usuario..	50
Tabla 2.66 Tiempos totales de entrega por iteraciones.....	51
Tabla 2.67: Diccionario de datos de la entidad alcance.....	55
Tabla 2.68: Diccionario de datos de la entidad archivos.....	56
Tabla 2.69: Diccionario de datos de la entidad contratista.....	56
Tabla 2.70: Diccionario de datos de la entidad contrato.....	56
Tabla 2.71: Diccionario de datos de la entidad disciplinas.....	57
Tabla 2.72: Diccionario de datos de la entidad equipos.....	57
Tabla 2.73: Diccionario de datos de la entidad proyecto.....	57
Tabla 2.74: Diccionario de datos de la entidad proy_disciplina.....	58
Tabla 2.75: Diccionario de datos de la entidad sistemas.....	58
Tabla 2.76: Diccionario de datos de la entidad usuarios.....	59
Tabla 2.77: Diccionario de datos de la entidad usuario_activo.....	59
Tabla 2.78: Descripción de iconos, botones y enlaces.....	77
Tabla 4.1: Prueba de aceptación 1.....	112
Tabla 4.2: Prueba de aceptación 2.....	113

Tabla 4.41: Matriz de evaluación de software -SUS (System Usability Scale).....	115
Tabla 4.42: Tabla de costos.....	117

ÍNDICE DE FIGURAS

Figura 1.1: Diferencia entre la programación del lado del servidor y la del lado del cliente.....	13
Figura 2.1: Diagrama de flujo del desarrollo de un proyecto de ingeniería de TECNIE.....	29
Figura 2.2: Diagrama de flujo del sistema de control de documentación en TECNIE.....	30
Figura 2.3: Arquitectura cliente-servidor.....	52
Figura 2.4: Diagrama de clases de SGI.....	53
Figura 2.5: Diagrama de base de datos de SGI.....	54
Figura 2.6: Spike de interfaz inicial.....	62
Figura 2.7: Spike de interfaz acceso.....	62
Figura 2.8: Spike de menú principal.....	62
Figura 2.9: Spike de interfaz de una vista de consulta 1.....	63
Figura 2.10: Spike de interfaz de una vista de consulta 2.....	63
Figura 2.11: Spike de interfaz de una vista de consulta 3.....	64
Figura 2.12: Spike de módulo de edición.....	64
Figura 2.13: Spike de interfaz de módulo primario (Proyecto) de edición de registros.....	64
Figura 2.14: Spike de interfaz de módulo secundario (Archivos) de edición de registros.....	65
Figura 2.15: Spike de interfaz de edición de usuarios.....	65
Figura 2.16: Spike de interfaz de módulo de logs.....	66
Figura 2.17: Cabecera de plantilla.....	67
Figura 2.18: Pantalla inicial.....	67
Figura 2.19: Ingreso de credenciales.....	68
Figura 2.20: Cabecera de administrador.....	69
Figura 2.21: Cabecera de editor.....	69
Figura 2.22: Cabecera de lector.....	69
Figura 2.23: Menú principal.....	70
Figura 2.24: Vista de consulta 1.....	70

Figura 2.25: Vista de consulta 2.....	71
Figura 2.26: Vista de consulta 3 – layout.....	72
Figura 2.27: Modulo edición.....	73
Figura 2.28: Cabecera de modulo edición.....	73
Figura 2.29: Módulo de edición primario “proyectos”.....	74
Figura 2.30: Módulo de edición secundario “archivos”.....	74
Figura 2.31: Módulo de administrador.....	75
Figura 2.32: Módulo de registro de ediciones.....	75
Figura 2.33: Mensaje de aviso de ingreso al sistema.....	77
Figura 2.34: Mensaje de aviso de inserción correcta.....	78
Figura 2.35: Mensaje de aviso de acción no permitida.....	78
Figura 2.36: Mensaje de aviso con opción a elección.....	78
Figura 2.37: Mensaje de aviso con alerta.....	79
Figura 2.38: Mensaje de aviso de cierre de sesión por inactividad.....	79
Figura 2.39: Código de la clase seguridad.....	81
Figura 2.40: Código de la función AccederModuloUsuario().....	82
Figura 2.41: Código de la clase conexion.....	83
Figura 2.42: Código de la clase registros.....	84
Figura 2.43: Código de la función nuevoRegistro().....	85
Figura 2.44: Código de llamado a la función nuevoRegistro().....	86
Figura 2.45: Código de llamado a la función agregarRegistros().....	86
Figura 2.46: Código de llamado a la función editarRegistros().....	87
Figura 2.47: Código de llamado a la función actualizarRegistros().....	88
Figura 2.48: Código de llamado a la función eliminarRegistros().....	89
Figura 2.49: Código de llamado a la función listarRegistros().....	90
Figura 2.50: Código de la clase ingresodatos().....	91
Figura 2.51: Código de llamado a la función agregarRegistros() para la agregación de registros secundarios.....	92
Figura 2.52: Código de la función agregarRegistros().....	93
Figura 2.53: Código CSS – Hojas de estilo.....	94
Figura 2.54: Código JavaScript – Funciones.....	94
Figura 3.1: Infraestructura de red de TECNIE.....	99
Figura 3.2: Creación de reglas de port forwarding.....	101

Figura 3.3: Configuración para restringir acceso a la base de datos desde PHPMyAdmin.....	103
Figura 3.4: Ejemplo de código de implementación de trigger.....	105
Figura 3.5: Módulo de logs.....	106
Figura 3.6: Código de archivo batch.....	106
Figura 3.7: Exportar código MySQL desde PHPMyAdmin.....	107
Figura 3.8: Archivo de texto plano, código sql.....	108

RESUMEN

Este proyecto tiene como propósito el diseño e implementación de una plataforma Web para gestionar y optimizar el almacenamiento y control de información técnica, legal-contractual y financiera generada en cada proyecto desarrollado por la empresa TECNIE.

En el primer capítulo inicialmente se presentarán los sustentos teóricos que facilitaran el desarrollo del proyecto para cumplir el objetivo planteado. Están incluidos fundamentos de metodologías de desarrollo Web, plataformas y servidores Web, gestores de contenido, lenguajes de programación y desarrollo del lado del servidor y cliente en ambiente Web, bases de datos relacionales y sistemas de gestión de bases de datos para así poder seleccionar las herramientas, opciones y tecnologías más adecuadas.

En el segundo capítulo se realizará un breve análisis de la situación actual de la empresa TECNIE sobre la gestión de información de los proyectos en los que ha trabajado, su giro de negocio y su forma de trabajo. Se establecerán antecedentes del problema a resolver y el planteamiento de la solución, se iniciará el estudio preliminar del manejo, control y gestión de información/documentación de la empresa TECNIE, luego se realizará la selección de la metodología de desarrollo adecuada, se definirá el equipo de trabajo desarrollador-cliente, se definirán alcances, requerimientos de la plataforma Web y roles de usuario, se planificarán las fechas de entrega y se realizará la codificación con sus respectivas pruebas unitarias, durante la entrega de cada fase de desarrollo se aplicaran las correcciones necesarias.

En el tercer capítulo se implementarán las seguridades, gestión de transacciones y respaldos necesarios a realizarse para resguardo de la integridad de la información de la plataforma Web implementada.

En el cuarto capítulo se realizará pruebas de aceptación con el cliente y se hará el análisis de resultados y costos del aplicativo Web. Finalmente en el quinto capítulo se presentarán las conclusiones y recomendaciones.

PRESENTACIÓN

En la actualidad durante el desarrollo y ejecución de un proyecto en empresas de ingeniería y fiscalización se genera un gran conocimiento e información. No obstante, la mayor parte de esto no queda capturado por la misma empresa, sino que queda simplemente de experiencia para los involucrados directos perdiéndose así el potencial de poder re-utilizarlos y compartirlos para otros proyectos futuros que tengan actividades similares. Teniendo en cuenta esto se puede evidenciar claramente que la información y conocimientos generados en cada proyecto son realmente valiosos, y el poder capturarlos y usarlos de una manera organizada, para no repetir errores y mejorar procesos es una gran ventaja competitiva frente a sus similares.

Consecuentemente gracias al desarrollo, crecimiento, facilidades y potencialidad de las nuevas tecnologías y servicios Web que trabajan conjuntamente con una red de datos o internet es posible la implementación de tal solución que permite la captura, gestión y control de tal información.

Es así que se plantea y pormenoriza el desarrollo de un aplicativo o plataforma Web de acceso local y remoto para gestionar la información generada de proyectos de ingeniería y fiscalización realizados por la empresa TECNIE.

CAPÍTULO 1

FUNDAMENTOS TEÓRICOS

1.1 METODOLOGÍAS DE DESARROLLO WEB

La metodología de desarrollo Web se define como los procedimientos que permiten planificar y estructurar de manera simple los procesos y etapas de desarrollo de software Web, existen dos tipos de metodologías de desarrollo: tradicionales y ágiles.

1.1.1 METODOLOGÍAS DE DESARROLLO TRADICIONAL^[1]

Se definen como aquellas metodologías que buscan elaborar software más previsible y eficiente, para obtener dichos resultados imponen un disciplina de trabajo y planificación sobre el desarrollo del software, debido a esto poseen un alto tiempo de ejecución y costo por la gran cantidad de miembros del equipo de trabajo ya que en dicha metodología se asumen todos los detalles desde un inicio mediante una planificación exhaustiva. Generalmente se usan para grandes proyectos.

Por otra parte este tipo de metodología no es recomendable ser aplicada a proyectos donde el entorno es variable a corto plazo, debido a las dificultades para implementar cambios.

1.1.2 METODOLOGÍAS DE DESARROLLO ÁGIL

Son aquellas metodologías que buscan conseguir que el desarrollo de software sea rápido y que responda fácilmente a cambios que puedan surgir en el tránscurso del proyecto o su implementación, permiten la interacción directa con el cliente lo que facilita enormemente un desarrollo incremental del software de alta calidad, el número de miembros del equipo de desarrollo es bastante reducido, por lo que solo se necesitan: el cliente, uno o dos desarrolladores y un jefe de proyecto.

Por lo general este tipo de metodologías son aplicadas sobre proyectos que están previstos para un lapso de tiempo relativamente pequeño, dentro de este conjunto

de metodologías de desarrollo de software una de las más destacadas es XP(Extreme Programming), la cual tiene como objetivo fundamental una retroalimentación continua entre el usuario y el desarrollador, teniendo como premisa que los cambios en los requisitos son un aspecto fundamental en el desarrollo, lo que hace que este método sea más adaptativo que previsorio a nivel global.

1.1.3 COMPARACIÓN DE METODOLOGÍAS TRADICIONALES Y ÁGILES

En la Tabla 1.1 mostrada a continuación se definen las principales diferencias entre ambas metodologías.

METODOLOGÍAS TRADICIONALES	METODOLOGÍAS ÁGILES
Se basan en normas y estándares para su desarrollo.	Se basan en reglas metodológicas informales provenientes del proceso de producción de código.
Su proceso es mucho más controlado.	Realiza los procesos con menos control.
Equipo de trabajo con amplio personal.	Equipo de trabajo reducido.
No existe una interacción directa entre el cliente y el desarrollo.	El cliente es parte del equipo de desarrollo.
Existe un contrato prefijado, con políticas que deben hacerse cumplir.	No existe contrato tradicional o al menos es bastante flexible.
Ofrecen resistencia a los cambios.	Especialmente preparadas para cambios durante el proyecto.
Se orienta a proyectos de larga realización.	Orientada a proyectos de corto desarrollo y con tiempo reducido.

Tabla 1.1: Diferencia entre metodologías tradicionales y ágiles^[2]

1.1.4 CARACTERÍSTICAS FUNDAMENTALES DE LA METODOLOGÍA XP

1.1.4.1 Simplicidad de código

El sistema a desarrollarse debe ser diseñado con la mayor simplicidad posible y dado el caso de que surja alguna complejidad esta deberá proceder a eliminarse inmediatamente buscando nuevas alternativas.

1.1.4.2 Refactorización del código

Implica que el programador o equipo de programación puede realizar una reestructuración en el sistema sin que sea necesario variar su funcionalidad o comportamiento, con el motivo de simplificar código y remover duplicados innecesarios.

1.1.4.3 Integración frecuente de desarrollador-usuario

Inclusión de un representante del cliente para trabajar conjuntamente en equipo con el/los desarrollador/es.

1.1.4.4 Corrección total de errores

Realización de verificación de todos los errores antes de añadir una nueva funcionalidad.

1.1.4.5 Pruebas unitarias continuas

Son pruebas o test previos del código implementado antes de su inclusión en el código principal y antes de que se le muestre al cliente, dichas pruebas son realizadas por el programador.^[3]

1.1.4.6 Roles XP

1.1.4.6.1 Programador

Es el encargado de construir el sistema y de tomar decisiones, el programador diseña, programa y realiza pruebas.

1.1.4.6.2 Jefe de proyecto

Es aquella persona que planifica las reuniones, igualmente asegura las condiciones óptimas para el desarrollo del proyecto, así como las facilidades para que se realice correctamente.

1.1.4.6.3 Cliente^[4]

Es aquella persona que determina cuando y que se debe elaborar, también este realiza las respectivas pruebas de aceptación, pudiendo el jefe del proyecto también estar incluido.

1.1.4.7 Ciclo de vida de la metodología XP

1.1.4.7.1 Etapa de exploración

Etapa inicial en la que el cliente hace un planteamiento generalizado de las historias de usuario en donde el desarrollador se empieza a familiarizar con las herramientas de desarrollo que se van a utilizar para la codificación y respectiva implementación del proyecto.

1.1.4.7.2 Etapa de planificación de entrega^[5]

Etapa en la que el cliente procede a establecer las prioridades a las historias de usuarios, el desarrollador verifica dichas prioridades y estima el esfuerzo que le conlleva cumplir cada una de ellas. Luego de esto se realiza un cronograma de entrega con el cliente. El punto es la medida de estimación de esfuerzo relacionado al tiempo usado para la implementación y compleción de las historias de usuario. Un punto generalmente suele equivaler a una semana ideal de programación, sin embargo, el desarrollador puede optar por establecer reglas adicionales para la estimación.

1.1.4.7.3 Etapa de Iteraciones

Son las etapas de implementación y compleción de un grupo de historias de usuario, que son realizadas acorde a la priorización que el usuario o cliente objetivo establece. Usualmente se desarrollan entre tres o cuatro iteraciones dependiendo del proyecto a implementar, con el objetivo fundamental de realizarlas antes de entregar el aplicativo al cliente para las pruebas de aceptación. Inicialmente se puede establecer la arquitectura del sistema que debe ser usada para el desarrollo del proyecto.

1.1.4.7.4 Etapa de producción

Dicha etapa necesita que sean implementadas pruebas adicionales y revisiones de rendimiento antes de que el sistema sea ejecutado por el cliente, en su entorno. Igualmente en esta etapa se incluyen nuevas características debido a cambios surgidos.

1.1.4.7.5 Finalización del sistema^[6]

Al final cuando la etapa de producción ha terminado y el sistema ya está en funcionamiento, surgen nuevas necesidades tales como rendimiento y confiabilidad del sistema, las cuales deben ser solventadas con el objetivo de entregar un buen producto.

1.2 PLATAFORMAS Y SERVIDORES WEB

1.2.1 PLATAFORMA WEB

1.2.1.1 Definición de plataforma Web

Es una colección de tecnologías Web sobre la cual funciona un aplicativo que procesa contenido público delimitado.

1.2.1.2 Plataforma^[7]

Se refiere a un sistema base que permite el funcionamiento de determinados módulos compatibles de hardware y software, estos sistemas se definen mediante estándares que permiten determinar una arquitectura de hardware y una plataforma de desarrollo de software.

1.2.1.3 Plataforma de desarrollo de software

Al desarrollar un programa o aplicación sobre una plataforma de desarrollo, su funcionamiento depende directamente del sistema operativo en sí, es decir que es necesario para su ejecución por que usa los recursos del mismo, existe la ventaja

de su ejecución ya que puede correr en cualquier navegador (de cada distribución), lo que vuelve a este tipo de programas más versátiles.

Actualmente existen numerosas plataformas que tienen como objetivo desarrollar y ejecutar sitios Web, la justificación de su elección será dependiente de los siguientes factores: soporte, velocidad, costos, flexibilidad.

Entre las plataformas para desarrollo Web más comunes se encuentran las siguientes:

- LAMP(Linux, Apache, MySQL y PHP)
- WAMP(Windows, Apache, MySQL, PHP)
- XAMP(X=Cualquier Sistema Operativo, Apache, MySQL, PHP)
- WXCF(Windows, X= Apache o IIS , Coldfusion)
- WIMA(Windows, IIS, MS SQL Server, ASP.NET)^[8]

1.2.1.3.1 LAMP^[9]

Es una de las plataformas para desarrollo Web más usadas, la cual usa Linux como sistema operativo, Apache como servidor Web, MySQL como sistema de administración de base de datos relacionales y PHP como lenguaje orientado a objetos.

Ventajas:

- Alta disponibilidad y bajos costos.
- Rapidez y un mejor tiempo efectivo de actividad.
- Facilidad para colocar instrucciones en el código.
- Posee un servidor de base de datos relacional bastante rápido para consultas de lectura.

Desventajas:

- Bajo rendimiento para sitios Web demasiado grandes y con un gran volumen de inserción de datos.

1.2.1.3.2 WAMP^[10]

Es una plataforma para desarrollo Web que usa Windows como sistema operativo y un conjunto de herramientas y tecnologías de desarrollo libre para ambientes Web tales como Apache (Servidor Web). MySQL se define como sistema de administración de base de datos relacionales y PHP como lenguaje orientado a objetos.

Por lo general los programadores que poseen un sistema operativo en Windows prueban su aplicativo desarrollado en la plataforma WAMP y luego lo suben a una plataforma LAMP.

Ventajas:

- Idealizado para usarse como servidor de desarrollo.
- Instalación bastante fácil del software (conjunto de aplicaciones).

Desventajas:

- Existen ciertas incompatibilidades al momento de instalar el servidor Web Apache, ya que por defecto el servidor Web que corre sobre windows es IIS(Internet Information Services).

1.2.1.3.3 XAMP^[11]

Es una plataforma para desarrollo Web basada en herramientas y tecnologías libres, dicha solución incluye y configura de manera rápida y compacta PHP, MySQL y Apache. Puede ser instalada en cualquiera de los siguientes sistemas operativos: Windows, Linux, Sun Solaris y Mac OS X.

1.2.1.3.4 WXCF^[12]

Esta plataforma para desarrollo Web usa a Coldfusion como lenguaje de programación orientado a objetos la cual funciona mediante etiquetas similares al lenguaje HTML, además puede instalarse en Windows, Mac OS X, Linux y Solaris..

Así mismo permite trabajar con bases de datos como Sybase, Oracle, MySQL, SQL Server, o Access mediante el standard SQL.

Ventajas:

- Coldfusion permite elegir el servidor Web sobre el que va a correr (Apache o IIS).
- Permite una fácil programación a través de etiquetas o tags (similar a la programación en HTML).
- Es integrable con Java.
- Posee extensas opciones para la configuración del servidor (en modo gráfico).

Desventajas:

- Muy pocos proveedores de hosting ofrecen este tipo de plataforma Web.
- Servidor inestable, soporta poca carga.
- Alto coste por su adquisición.

1.2.1.3.5 WISA^[13]

Es una plataformas para desarrollo Web formada por un paquete de herramientas y tecnologías de Microsoft, para creación y ejecución de software Web. El paquete consiste en Windows como sistema operativo, IIS (Internet Information Services) como servidor de archivos o servidor Web, SQL Server como software de base de datos y ASP.NET u otra tecnología .NET como lenguaje de programación Web.

Ventajas:

- Brinda un poderoso entorno gráfico para su administración y manipulación al momento de la programación.
- Permite brindar a múltiples usuarios cantidades altas de información simultánea.

- IIS puede resultar fácil de configurar.
- Amplio soporte ofrecido por Microsoft.
- ASP.NET puede resultar fácil de manejar a la hora de desarrollar páginas Web.

Desventajas:

- El servidor IIS sólo funciona bajo Windows.
- El servidor SQL Server únicamente funciona en el sistema operativo Windows
- Existen algunas limitaciones en IIS en las versiones que no son de tipo Servidor.
- Alto costo por adquisición de licencias.

1.2.2 SERVIDORES WEB

1.2.2.1 Internet^[14]

Se define como Internet al conjunto de redes interconectadas entre sí, que permite la comunicación entre computadores por medio del protocolo TCP/IP.

1.2.2.2 El servicio Web^[15]

Se define al servicio Web como el conjunto de aplicaciones que están distribuidas por toda la red en forma de documentos hipertextuales que permiten intercambiar datos para ofrecer servicios.

1.2.2.2.1 Hipertexto

Es aquel documento multimedia que se enlaza con otros documentos o recursos, por medio de una plataforma informática. El mismo tiene la función de integrar todas las tecnologías de la información, incorporando y generando relaciones estructurales.

1.2.2.3 Relación cliente/servidor

Todos servicio ofrecido en el Internet especialmente los de tipo Web, se basan en una relación cliente-servidor. En Internet se encuentran dos tipos de equipos conectados:

1.2.2.3.1 Servidores

Ordenadores o equipos generalmente con características físicas de hardware bastante robustas que ofrecen una gran variedad de servicios a otros equipos conectados, y que comúnmente siempre suelen estar disponibles en la red interna o externa (de manera remota). En un entorno Web, en estos ordenadores se encuentran alojadas las páginas Web. Normalmente poseen un direccionamiento de IP fijo o estático.

1.2.2.3.2 Clientes^[16]

Equipos de usuarios individuales usados para conectarse a la red, dichos equipos realizan peticiones de servicios a los servidores. Usualmente mantienen una presencia física no permanente en la red. A menudo estos equipos poseen dirección IP variable la cual es asignada por el proveedor de internet y se mantiene solo durante su conexión, es decir que es dinámica.

1.2.2.4 Servidor Web^[17]

Se define como el aplicativo o programa que procesa peticiones de sus clientes usando el protocolo HTTP y entrega respuestas y resultados en forma de páginas Web.

1.2.2.4.1 Programa^[18]

Son instrucciones o enunciados que ejecuta una computadora y que permiten realizar determinadas actividades. Los mismos son realizados mediante lenguajes de programación.

En un ámbito Web existen dos tipos de programa: los programas cliente y los programas servidor, a continuación se procede a detallar cada uno de ellos:

Programa Cliente: Se define como el software necesario en el equipo (cliente) que permite tener acceso al respectivo servicio. Se pueden citar varios ejemplos como Google Chrome, Internet Explorer o Mozilla Firefox que serían clientes Web y por otro lado Microsoft Outlook sería un cliente de correo electrónico.

Programa servidor: Es el programa que se está ejecutado en el servidor para que este pueda ofrecer su servicio. En la práctica un documento HTML que solo está almacenado en el equipo del cliente no es suficiente para que sea accesible como página Web por Internet, es necesario que en ese equipo este corriendo una aplicación que sirva como servidor Web. Se pueden citar como ejemplo a Apache, que es el servidor Web libre más conocido y a IIS (Internet Information Services) que está disponible en equipos que usan sistema operativo Windows de la firma Microsoft.

1.3 GESTORES DE CONTENIDO^[19]

Se denomina a un gestor de contenidos Web como el software que posibilita el desarrollo de un sitio Web y gestión de su contenido incluyendo herramientas para edición de formatos, estructura y presentación sin la necesidad de poseer conocimientos técnicos.

Su funcionamiento puede resumirse de la siguiente forma:

- Un sitio Web se encuentra conformado de manera jerárquica por un conjunto de páginas organizadas.
- Cada una de estas páginas posee un diseño determinado que define su estructura y formato.
- Cada página se encuentra asociada a un campo específico de la base de datos en la cual se depositan y almacenan los contenidos.

1.4 LENGUAJE DE PROGRAMACIÓN^[20]

Es un lenguaje formal y estructurado diseñado para colocar un conjunto de instrucciones o procesos para controlar el comportamiento físico y lógico de una máquina. Su estructura, significado de elementos y expresiones están definidas por símbolos, reglas sintácticas y semánticas. Teniendo en cuenta las características mencionadas, el proceso por el cual se escribe, prueba, depura y compila el código de un programa/código recibe el nombre de programación.

1.4.1 TIPOS DE LENGUAJES DE PROGRAMACIÓN DE ALTO NIVEL

1.4.1.1 Lenguajes compilados^[21]

Es aquel lenguaje en donde el código fuente pasa por un proceso llamado compilación generando un código objeto el mismo que se enlaza con otros módulos de código objeto, para finalmente crear un archivo ejecutable. Dicho archivo contiene todas las instrucciones del mismo pero en el formato entendible por la máquina.

1.4.1.2 Lenguajes interpretados^[22]

Los leguajes interpretados a diferencia de los compilados poseen una aplicación llamada interprete que se encarga de ir traduciendo en un orden secuencial el código el código o instrucciones, de modo que al ejecutan al mismo tiempo que lo hace el programa, por lo que si se llegase a un punto en el que exista alguna instrucción errónea en el código, el programa se seguirá ejecutando pero sin embargo ya habrá podido ejecutar todas las instrucciones o sentencias anteriores.

1.5 DESARROLLO DEL LADO DEL SERVIDOR Y CLIENTE EN AMBIENTES WEB

En ambientes Web el desarrollo de este tipo de programas: cliente y servidor, mostrados en la Figura 1.1 funcionan análogamente a un emisor y receptor de

código. En donde la programación del servidor ejecuta el programa y envía el código HTML al cliente quien lo interpreta y procesa con ayuda del navegador.



Figura 1.1: Diferencia entre la programación del lado del servidor y la del lado del cliente^[23]

1.5.1 DESARROLLO DEL LADO DEL SERVIDOR^[24]

La programación del lado del servidor consiste en procesar una petición realizada por un usuario mediante de la interpretación de un script en el servidor Web para la generación de páginas HTML en el lado del cliente que se generan dinámicamente como respuesta, facilitando el acceso a las bases de datos. Los lenguajes comúnmente usados para este propósito pueden ser PHP o ASP.

1.5.2 DESARROLLO DEL LADO DEL CLIENTE^[25]

Son programas que residen junto a la página Web y que pueden ser procesados y ejecutados por el cliente directamente en el navegador sin la necesidad de ser enviados al servidor. Entre estos lenguajes se encuentran: Html, Java, JavaScript, VBScript.

1.5.3 DESARROLLO DEL LADO DEL CLIENTE VS SERVIDOR^[26]

Cada tipo de lenguaje de desarrollo y programación ya sea de lado del cliente o servidor posee determinadas ventajas y desventajas, se debe recalcar igualmente

que ambos tipos de desarrollo se pueden generar e incorporar una misma página por ejemplo la inclusión de código en PHP para que sean ejecutados por el servidor y scripts en Html, JavaScript o VBScript para se ejecuten por el cliente (en su navegador).

A continuación en la Tabla 1.2 se presenta un cuadro comparativo de las características de desarrollo y programación entre ambos lenguajes: cliente y servidor.

Programación del lado del cliente	Programación del lado del servidor
Los programas residen en el servidor pero se ejecutan en el cliente.	Los programas residen en el servidor y son ejecutados por mismo.
Se procesa directamente en el equipo (navegador) del cliente.	El trabajo recae sobre los servidores pudiendo llegar a sobrecargarse.
La ejecución del programa requiere una transmisión por la red del código necesario para ello.	Al cliente solo se le transfiere el resultado de la ejecución del programa.
Las respuestas a las acciones de los usuarios sobre el programa pueden ser invocadas sin necesidad de realizar transmisiones por la red.	Una vez enviada al usuario la respuesta del programa, cualquier petición adicional del cliente requiere una nueva conexión con el servidor y la ejecución en él de un nuevo programa.
Para la correcta ejecución del programa se requiere que el cliente tenga instalados programas o plug-ins adecuados.	En los equipos de los clientes no se necesita ningún software especial, todo lo necesario debe estar instalado en el servidor.
Si en un cliente no está instalado alguno de los programas intérpretes o plug-ins, la página no se ejecutará correctamente.	Todos los clientes podrán visualizar correctamente la página.
Al transferirse el código, el cliente tiene acceso a dicho código y puede obtener a partir de él información que pueda resultar comprometida.	El código fuente permanece en el servidor, se conserva su privacidad y los clientes no tienen acceso a él.
Se pueden integrar los programas en las páginas alojadas en cualquier servidor Web.	La mayoría de los servicios de alojamiento gratuito de páginas no admiten este tipo de programación.

Tabla 1.2: Diferencias entre la programación del lado del cliente y del lado del servidor^[27]

1.5.3.1 Lenguajes de programación del lado del servidor^{[28][29]}

1.5.3.1.1 PHP

Es un lenguaje de programación interpretado de código abierto orientado al servidor, su acrónimo significa PHP Hypertext Pre-processor.

Entre sus principales características se puede destacar su potencia, robustez, versatilidad y modularidad. Puede ser desplegado en la mayoría de los servidores Web y es considerado multiplataforma. Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Microsoft SQL Server, SQLite entre otros.

En su gran mayoría este lenguaje está orientado a facilitar la creación de sitios Web mediante scripts.

Su modo de funcionamiento opera de la siguiente manera: al momento en el que el cliente realiza una petición al servidor de una página Web, el servidor ejecuta el intérprete de PHP. El mismo hace el procesamiento del script que genera el contenido de manera dinámica. Finalmente este resultado se envía mediante el intérprete al servidor, que a su vez lo va a enviar al cliente.

1.5.3.1.2 ASP: Páginas de Servidor Activas^[30]

ASP (Active Server Pages), es un lenguaje de programación de servidores para generar páginas Web de manera dinámica, para su funcionamiento requiere de un servidor Web de Microsoft llamado “IIS - Internet Information Server”. Se conocen algunas versiones de este lenguaje las 1.0, 2.0, 3.0 y .NET.

El lenguaje ASP.NET es un lenguaje que permite separar en las páginas webs la parte de diseño que contiene la página de su funcionalidad, de modo que no se interviene para nada el código HTML. Así el trabajo de los diseñadores y programadores es mucho más sencillo, cada quien hace su labor sin incidir en el trabajo del otro. Usa como lenguajes de programación el VBScript y el Jscript.

Su modo de funcionamiento se da de la siguiente manera: un cliente realiza una petición de página ASP, dicha petición es interpretada por el servidor y este envía una página Web HTML al cliente. El cliente nunca llega a ver el código ASP, únicamente ve el resultado de la interpretación del mismo en código HTML.

1.5.3.1.3 JSP: Páginas de Servidor Java^[31]

Es una tecnología orientada a crear páginas Web con programación Java encargándose así de facilitar el desarrollo de sitios Web. En su programación se puede incorporar contenido dinámico en sitios Web usando código Java embebido a mediante etiquetas especiales.

Dichas páginas JSP son archivos de texto con extensión .jsp contenedoras de etiquetas HTML, junto con código Java embebido.

Su modo de funcionamiento se da de la siguiente manera: En el momento en el que se hace una petición de una página JSP, la parte HTML es procesada en el cliente, no obstante, el código Java es ejecutado al momento de recibir la petición y el contenido dinámico que se genera por aquel código es insertado en la página antes de devolverla al cliente.

1.5.3.2 Lenguajes de programación del lado del cliente

1.5.3.2.1 Java

Es un lenguaje de programación que requiere un proceso de compilación. El código compilado generado se lo puede integrar en la página Web para que sea ejecutado por el cliente. Fue desarrollado por la empresa Sun Microsystems y posteriormente fue comprado por Oracle.

Con Java es posible la creación dos tipos de programas:

- **Applets:** Son programas que están integrados en las páginas Web y residen en el servidor, los cuales a su vez son ejecutados por el cliente mediante una JVM (Java Virtual Machine) que el navegador tiene configurada como

extensión, el código fuente no está incrustado directamente en el archivo HTML, sino que únicamente se agrega como un código binario resultado de una compilación que tiene el nombre de JBC (Java Byte Code) lo que permite proteger el código fuente.

- **Aplicaciones^[32]:** Son programas autónomos e independientes que se ejecutan en cualquier computadora. Se compilan de manera similar a los applets y para ser ejecutados necesitan de un intérprete o código compilado ejecutable.

Entre sus principales características están las siguientes:

- Es orientado a objetos.
- Brinda prestaciones multimedia.
- Posee clases de objetos para la generación de interfaces gráficas de usuario.
- Sintaxis similar a C++.
- Es simple, robusto y seguro.

1.5.3.2.2 VBScript

Es un lenguaje interpretado permite integrar programas directamente en el código HTML. Puede ser usado tanto en el lado del cliente como en el del servidor para la generación de páginas ASP.

Este lenguaje fue desarrollado por Microsoft su sintaxis es bastante similar a Visual Basic para Aplicaciones, es una herramienta muy efectiva y fácil usar para crear páginas Web interactivas, tiene la desventaja de solo funcionar con el navegador Web de Windows “Internet Explorer”.

1.5.3.2.3 Javascript

Es un lenguaje de programación interpretado comúnmente usado para crear mini-aplicaciones encargados de realizar determinadas acciones en un ámbito de página

Web, tales como la creación de efectos especiales y contenidos dinámicos(elementos con movimiento, cambios de color...etc.) en páginas y la interacción con el usuario de acuerdo a las acciones que realice, su carga de procesamiento se da a nivel del navegador del equipo del cliente siendo este su mayor recurso, es bastante compatible y aceptado a nivel general. Se suele usar luego de haber implementado el código HTML.

1.6 BASE DE DATOS RELACIONALES

1.6.1 BASE DE DATOS^[33]

Es el conjunto de datos que pertenecen a un mismo contexto y se almacenan de manera sistemática para ser posteriormente usados.

1.6.1.1 Base de datos relacionales^{[34][35]}

Son aquellas bases de datos en donde los datos visibles al usuario se organizan mediante tablas de valores que varían con el tiempo, mismas que se interconectan entre sí de manera normalizada mediante determinadas operaciones, permitiendo que se establezcan relaciones entre los datos guardados en las tablas. La información puede ser recuperada o almacenada mediante consultas que ofrecen una gran flexibilidad y administración de la información que contienen.

Es decir que una base de datos relacional se considera como una colección de relaciones. En donde una relación representa una tabla, que viene a ser un conjunto de filas o registros, cada fila o registro es un conjunto de campos y cada campo representa un valor.

Entre las principales características están:

- No existe un orden específico de los campos y los mismos son creados de izquierda a derecha.
- La creación de registros no se da en un orden específico, de arriba hacia abajo.

- Cada uno de los campos tienen un valor único.
- Existe un campo de identificación único llamado clave primaria.

1.7 SISTEMAS DE GESTIÓN DE BASES DE DATOS^[36]

Un Sistema de Gestión de Bases de Datos (SGBD) es un software o programa que almacena, modifica y extrae información de una base de datos. Los SGBD igualmente implementan métodos con el fin de mantener la integridad de los datos, permitiendo así administrar el acceso de usuarios, gestionar su acceso concurrente y recuperar la información dado que el sistema se falle o se corrompa. Permite presentar la información de la base de datos en variados formatos.

Se suele acceder a sus datos mediante lenguajes de programación estructurados y lenguajes consulta propios de que simplifican la tarea de construcción de las aplicaciones.

1.7.1 MYSQL^[37]

Es un sistema de gestión de bases de datos relacional, multi-usuario y multi-hilo bastante rápido y robusto, bajo la licencia GNU GPL (código abierto). Usa como estándar de base de datos el lenguaje SQL (Structured Query Language).

Principales características de MYSQL:

- Se encuentra en una gran cantidad de plataformas y sistemas usados actualmente.
- Trabaja en modo cliente-servidor.
- Permite la elección del tipo de almacenamiento que tendrán los datos a ingresarse en la base de datos permitiendo una mayor velocidad de operación, capacidad, transacciones...etc.
- En la actualidad es uno de los sistemas de gestión de base de datos con más presencia en el medio a nivel global.
- Permite hacer búsquedas rápidas y realizar indexación con los campos de texto.

1.7.2 SQL SERVER

Microsoft SQL Server es un sistema para la gestión de bases de datos relacionales desarrollado por Microsoft. Sus lenguajes para consultas son T-SQL y ANSI SQL. Entre sus principales características se encuentran las siguientes:

- Soporte de transacciones.
- Soporta procedimientos almacenados, los cuales permiten un buen rendimiento.
- Incluye también un entorno gráfico de administración, que permite el uso de comandos DDL (Data Definition Language) y DML (Data Manipulation Data) gráficamente.
- Solo se puede ejecutar sobre el sistema operativo Windows
- Instalación bastante sencilla.
- Software privativo, es necesario pagar para adquirirlo.

1.7.3 ORACLE^[38]

Es un sistema de gestión de base de datos objeto-relacional (ORDBMS, Object-Relational Data Base Management System), desarrollado por Oracle, considerado uno de los sistemas de gestión de datos más completo y robusto que existe actualmente en el mercado, razón por la cual tiene dominado el mercado de servidores empresariales.

Entre sus principales características están las siguientes:

- Trabaja en un entorno cliente-servidor.
- Gran capacidad de usuarios conectados (conurrencia)
- Posee un muy alto rendimiento en lo que respecta a las transacciones.
- No es libre, la adquisición de sus licencias resulta sumamente costosa.
- Gran presencia en el medio, especialmente en la banca y entidades que manejan un extenso volumen de información.

CAPÍTULO 2

DISEÑO E IMPLEMENTACIÓN: PLATAFORMA WEB

2.1 ESTABLECIMIENTO DE LÍNEA BASE

2.1.1 HISTORIA^[39]

TECNIE C. LTDA es una corporación Ecuatoriana con base en Quito-Ecuador, que provee desde 1983 servicios de desarrollo de proyectos de clase mundial incluyendo el desarrollo del concepto, promoción, manejo de brokers en proyectos hacia inversionistas, consultoría especializada, ingeniería, diseño y gerencia de proyectos. La compañía trabaja prioritariamente en el upstream del sector petrolero, facilidades, oleoductos y generación de energía.

Actualmente cuenta con un personal de alrededor de 30 personas, con la matriz ubicada en Quito con 15 personas y una sucursal temporal en Guayaquil con 15 personas.

2.1.2 ANTECEDENTES DEL PROBLEMA

Actualmente en la empresa TECNIE, durante el desarrollo y ejecución de un proyecto se genera un gran conocimiento e información sea ésta legal, contractual, financiera, u otras áreas, en su gran mayoría la información es técnica lo que al final se traduce a la entrega de un Dossier de calidad/ingeniería/construcción por proyecto, sin embargo la mayor parte de esto no queda capturado por la misma empresa, sino que queda simplemente de experiencia para los involucrados directos perdiéndose así el potencial de poder utilizarlos y compartirlos para otros proyectos futuros que tengan actividades similares.

Teniendo en cuenta esto se puede evidenciar claramente que la información y conocimientos generados en cada proyecto son realmente valiosos, y el poder consultarlos y utilizarlos de una manera organizada, para no repetir errores y mejorar procesos, sería una ventaja competitiva frente a sus similares.

TECNIE en la actualidad no cuenta con un sistema centralizado de gestión de información que permita el manejo y administración eficiente de la misma referente a la consecución y entrega de sus proyectos.

Esta gestión de información en la empresa TECNIE es realizada empíricamente por el personal de control de documentación mediante una herramienta aplicativa (Microsoft Excel) que carece de las funcionalidades de un sistema centralizado y seguro para la gestión de información, refiriéndose más concretamente a un aplicativo Web que facilite tal labor.

2.1.3 SOLUCIÓN PROPUESTA

Se propone implementar una plataforma Web que permita centralizar el ingreso, edición y recuperación (con facilidad de búsqueda y acceso en base a criterios preestablecidos) de información perteneciente a la finalización de un proyecto durante el lapso en el que se desarrolle el dossier de calidad/ingeniería o construcción del mismo. Dicha solución tendrá características como la capacidad de verificación y auditoria del historial de información modificada por fecha y usuario, también permitirá gestionar la administración de permisos para usuarios a distintos niveles (administrador, editor de información y lector).

Adicionalmente la plataforma Web servirá como una biblioteca de consulta digital, para un cliente objetivo específico con el que esté trabajando la empresa, mediante un acceso usuario lector.

2.2 ESTUDIO PRELIMINAR DEL MANEJO, CONTROL Y GESTIÓN DE INFORMACIÓN/DOCUMENTACIÓN DE LA EMPRESA TECNIE

2.2.1 ANTECEDENTES

TECNIE es una empresa que se maneja principalmente mediante proyectos de ingeniería y fiscalización, tanto desde la planificación hasta la puesta en marcha. El procedimiento que sigue para desarrollar un proyecto de ingeniería es el siguiente:

2.2.1.1 Proceso de desarrollo de un proyecto en TECNIE:

Cada proyecto desarrollado e implementado por TECNIE sigue un esquema general ya estructurado, el cual está compuesto por las siguientes etapas:

- Licitación
- Oferta
- Contrato
- Ingeniería
- Construcción
- Comisionado
- Cierre del proyecto

Pudiendo variar el lapso de tiempo determinado para cada una de estas etapas.

2.2.1.1.1 Etapas del proyecto:

Licitación: El proyecto empieza con esta etapa, que involucra un concurso público realizado por una empresa (cliente-objetivo). En donde un determinado grupo de empresas es invitado por la empresa a un concurso de ofertas para la contratación y desarrollo de un proyecto de ingeniería.

Este concurso público va de la mano con una serie de documentos en donde se establecen todos los parámetros y condiciones que debe tener la oferta, entre los principales están los siguientes:

- a) Términos y condiciones del proceso de cotización de precios:
 - Objeto: Objetivo principal de la contratación.
 - Forma de presentación de la oferta: Requisitos para la presentación
 - Preguntas y aclaraciones
 - Evaluación técnica y económica: Condiciones sobre el proceso de evaluación.
 - Plazo de prestación del servicio: Plazo de contratación para el proyecto.

- Inhabilidades para contratar: Restricciones de contratación para empresas.
 - Información técnica adicional: Especificaciones técnicas adicionales.
- b) Preguntas y Respuestas: Interrogaciones a datos que no quedaron claros en la oferta.
- c) Matriz de evaluación económica:
- Presupuesto Referencial
- d) Matriz de evaluación técnica:
- Personal Propuesto
 - Aseguramiento de calidad
 - Sistema de avance de documentación

Oferta: Es el documento o conjunto de documentos resultante generado por el contratista (TECNIE) que se encuentra en el proceso de licitación para lograr la adjudicación del proyecto. En el cual se condensa toda la propuesta desarrollada conforme a términos, condiciones y formatos establecidos en el concurso.

Dicha oferta es enviada al contratante (cliente-objetivo) para que sea evaluada conjuntamente con otras y se tome a una decisión de elección.

Contrato: Una vez ganado el concurso se firma un acuerdo escrito entre el contratista (quien anteriormente era ofertante) y el contratante o cliente-objetivo, en donde se colocarán normas, leyes, procedimientos, plazos...etc. Respetando lo que indica la ley.

Ingeniería: Un proyecto de ingeniería se define como el planeamiento, organización, ejecución y control del total de actividades y recursos necesarios dentro de un tiempo determinado para lograr un objetivo específico. De la misma manera en su ejecución están presentes la aplicación de ciencias fisicomatemáticas, económicas e industriales.

Es en esta etapa en donde se da inicio al proyecto de ingeniería en sí. La misma consta de las siguientes sub-etapas:

- Ingeniería conceptual
- Ingeniería Básica
- Ingeniería de detalle

Ingeniería conceptual: Etapa inicial de un proyecto de ingeniería en donde se analiza la necesidad específica planteada por el contratante/cliente-objetivo. Se trazan diversas maneras de solución, que son evaluadas en base a criterios técnicos, económicos y sociales (Seguridad y medio ambiente).

De las alternativas existentes se elige la mejor opción, dicha elección puede ser realizada por el contratante/cliente-objetivo o conjuntamente con el contratista.

A continuación se colocan los aspectos principales definidos en esta etapa:

- Memoria descriptiva
- Viabilidad técnica
- Cronograma inicial del proyecto
- Costos de inversión
- Costos de mantenimiento
- Rentabilidad de inversión
- Previsión para aplicaciones futuras
- Diagramas de flujo de procesos
- Estudios técnicos básicos (estudios ambientales, estudios de riesgo)...etc.

Ingeniería básica: En esta etapa se profundiza la solución elegida previamente en la ingeniería conceptual, tomando en cuenta detalles como:

- Establecimiento de dimensiones generales del sistema
- Programación de etapas constructivas
- Calculo de presupuestos por ítems globales

El objetivo de la misma es la obtención de una idea muy clara acerca del proyecto de ingeniería a desarrollarse de modo que permita tomar decisiones efectivas.

A continuación se colocan los aspectos principales definidos en esta etapa:

- Revisión de diagramas de procesos, tubería e instrumentación (P&ID), unifilares...etc. Conforme a normas establecidas para cada área.
- Dimensionamiento de equipos.
- Selección de Hardware y Software a ser usado.
- Estudios técnicos y realización de cálculos (batimetría, mecánica, hidráulica, geotecnia, geofísica, oceanográfica, estudios ambientales, estudios de riesgos...etc.), dependiendo del tipo de proyecto a realizarse.
- Realización de cálculos preliminares e inventario inicial de equipos.
- Estudio para selección de proveedores.

Ingeniería de detalle: Es la última etapa de ingeniería, en donde se condensa todo el diseño final detallado del proyecto de ingeniería, tal información es usada para la construcción e instalación/montaje. Sin embargo la misma puede estar sujeta a modificaciones, ya que en ciertas ocasiones pueden ocurrir variaciones (por variación de costos estimados, falta de materiales...etc.).

A continuación se colocan los aspectos principales definidos en esta etapa:

- Lista y plano final de disposición de equipos.
- Diagramas definitivos de procesos, tubería e instrumentación (P&ID)
- Planos de rutas (tubería y cable).
- Cálculos definitivos y memorias detalladas de los sistemas mecánicos, hidráulicos y eléctricos.
- Procedimientos y protocolos
- Estudios técnicos.
- Cronograma de ejecución de obras
- Licencias y contratos.
- Hojas de datos...etc.

Construcción: Etapa del proyecto que se basa en la ingeniería de detalle, en la misma se adquiere (Compra de equipos y materiales - Procura), construye e instala todos los sistemas del proyecto de ingeniería conforme al diseño y las respectivas especificaciones técnicas desarrolladas previamente. Dicha etapa podría hacer variar ligeramente a la ingeniería de detalle.

Es en esta etapa en donde se empieza a realizar/generar el dossier de calidad/ingeniería/construcción. Este documento contiene un condensado de la siguiente documentación mencionada a continuación:

- Técnica:
 - Ingeniería: Planos AS built, Resumen general de toda la ingeniería.
 - Calidad: Registros de pruebas realizadas en la construcción.
 - Equipos: Manuales de instalación, operación y mantenimiento
- Financiera y legal-contractual:

Toda esta documentación se encuentra referenciada al desarrollo y puesta en marcha del proyecto de ingeniería y tiene como objetivo fundamental demostrar la calidad del producto entregado al contratante/cliente-objetivo, es decir que dicho proyecto de ingeniería fue desarrollado bajo todas las normas y parámetros correspondientes.

Se procede a colocar la información más relevante que posee un dossier de calidad/ingeniería/construcción:

- Ingeniería de detalle
- Manuales / Catálogos
- Certificados
- Planos As built
- Contratos
- Facturas
- Pruebas

- Procedimientos
- Mantenimiento
- Hojas de datos

Comisionado: Comprende la supervisión de los trabajos del grupo de pre-comisionado, comisionado y arranque adelantado por contratista de construcción y los diferentes representantes de los fabricantes de equipos.

Pre-comisionado: Sub-etapa orientada al chequeo y verificación aislada de equipos, tiene como objeto el aseguramiento de que la construcción e instalación fueron desarrolladas en base a documentos de ingeniería de detalle aprobados previamente. El personal especializado verificará los certificados de pruebas realizadas durante la instalación e identificará los puntos pendientes que requerirán ser corregidos por el contratista de construcción.

Dicha información será documentada y entregada para el comisionado.

Comisionado y arranque: Una vez terminada la etapa de pre-comisionado, se verifica y procede a autorizar las pruebas funcionales de sistemas completos definidas por las especificaciones de construcción y de los fabricantes de equipos. El grupo de comisionado atestiguará las pruebas funcionales llevadas a cabo por el contratista de construcción.

Finalmente el arranque será autorizado al completar las pruebas funcionales satisfactoriamente.

Cierre del proyecto: Etapa final en la cual se transfiere el proyecto al cliente-objetivo/contratante, reuniendo todos los documentos necesarios (entre ellos dossier de calidad/ingeniería/construcción: Ingeniería, Calidad y Equipos) para la entrega oficial del proyecto. Al final se emitirá un acta de entrega definitiva del proyecto, la cual deberá ser firmada por ambas partes y terminará el proyecto.

A continuación en la Figura 2.1 se muestra un diagrama de flujo de todas las etapas del desarrollo de un proyecto de ingeniería, indicando en cuál de ellas se empieza a desarrollar el dossier de calidad/ingeniería/construcción:

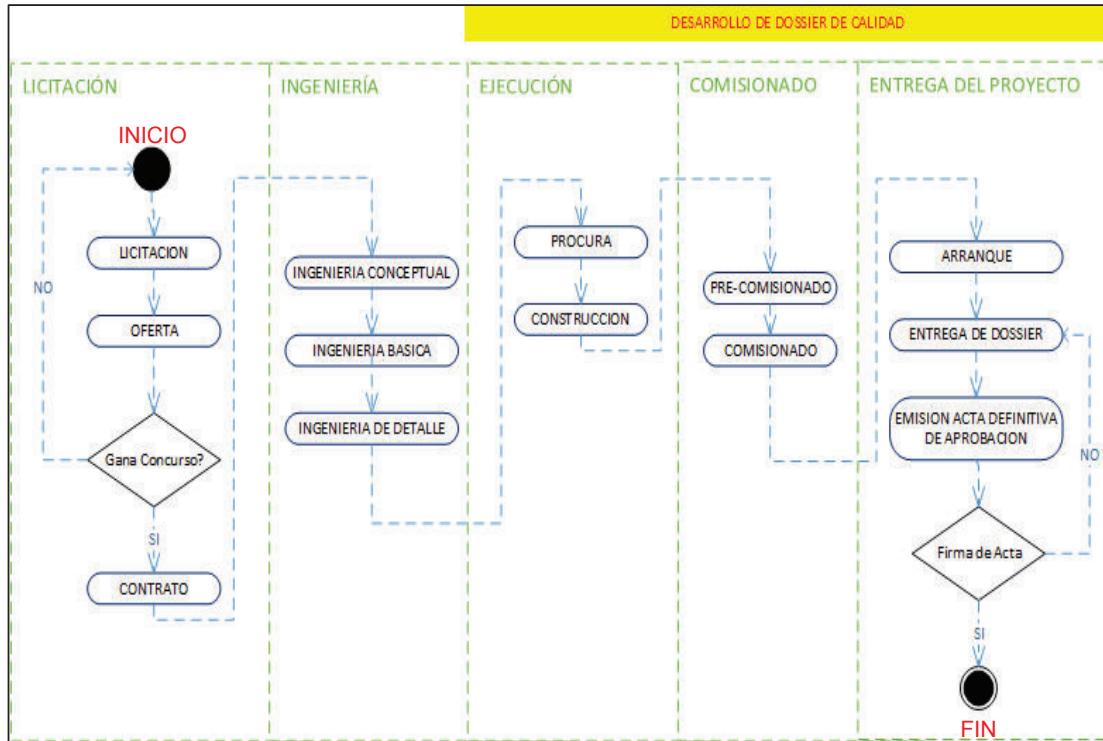


Figura 2.1: Diagrama de flujo del desarrollo de un proyecto de ingeniería de TECNIE

2.2.1.2 Sistema de control de documentación

Actualmente TECNIE mantiene un procedimiento de sistema de control de documentos debido a que el volumen de documentos generados durante el desarrollo de un proyecto de ingeniería es sumamente extenso, se hace necesaria la inclusión de una metodología de trabajo que permita controlar adecuadamente todo el flujo de documentación generada.

Dicha metodología deberá ser acatada por cada uno de los contratistas (empresas) que estén involucrados en el proyecto. El área encargada de coordinar que dicho sistema de control sea cumplido a cabalidad será el departamento de control de documentación de TECNIE.

A continuación en la Figura 2.2 se muestra un diagrama de flujo del proceso de control de documentación realizado en la empresa TECNIE:

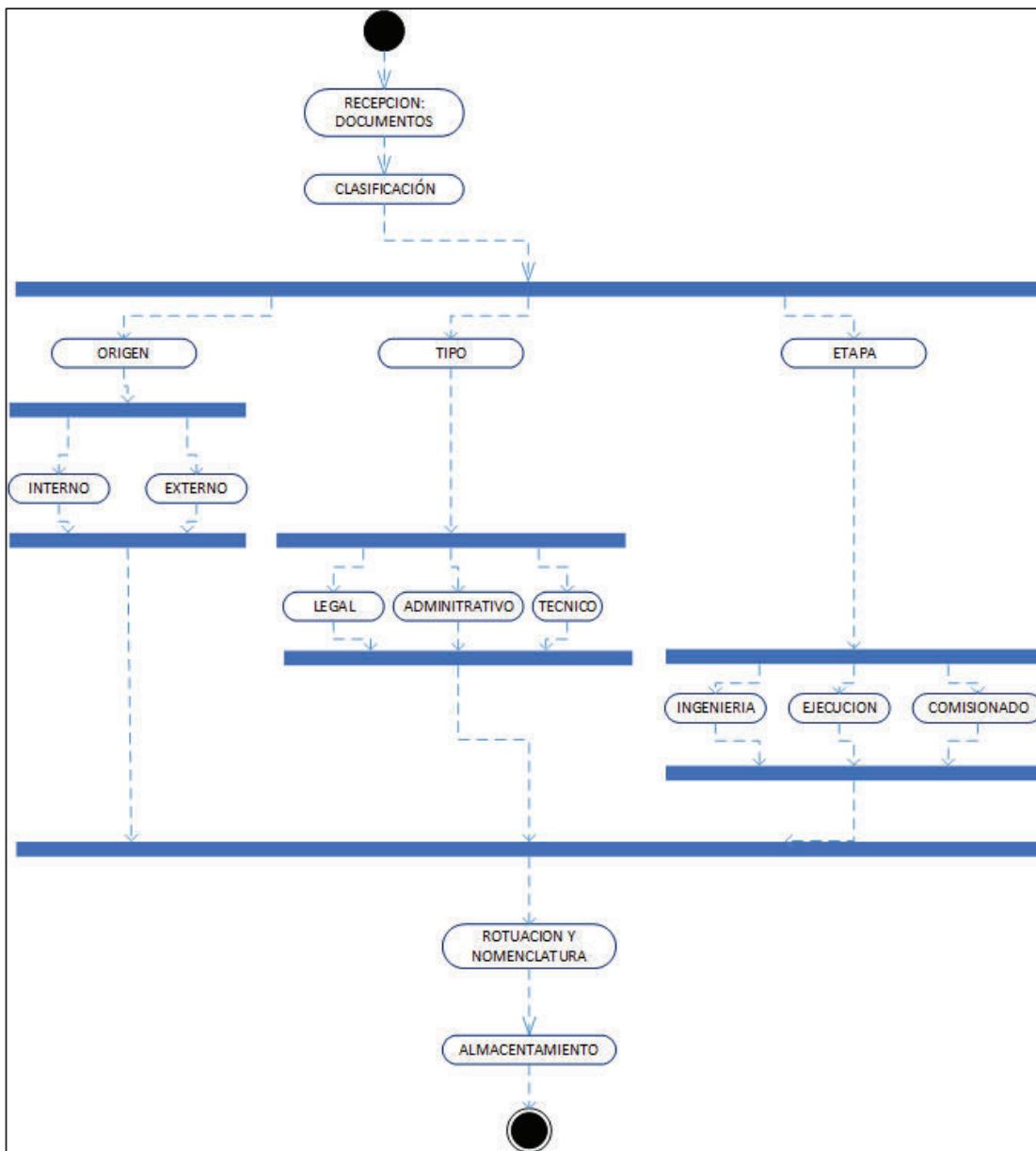


Figura 2.2: Diagrama de flujo del sistema de control de documentación en TECNIE

Para todos los documentos generados y recabados durante las etapas de desarrollo del proyecto de ingeniería o fiscalización se definirá:

- Una metodología de manejo y administración.
- Un sistema abreviado y lógico de identificación.

- Una clasificación de acuerdo a su fase o etapa.
- Un formato a utilizarse.

2.2.1.2.1 Tipo de documentación manejada

Documentación Interna: Información generada por TECNIE.

- Cartas Emitidas
- Documentos técnicos (Planos, Manuales...etc.)
- Emisión de Actas o minutos, Cronogramas, Histogramas, Informes,
- Instructivos y Listas de distribución.

Documentación Externa: Información generada fuera de TECNIE.

- Cartas recibidas
- Documentos técnicos (Planos, Manuales...etc.)
- Recepción de Actas o minutos, Cronogramas, Histogramas, Informes, Instructivos y Listas de distribución.
- Procedimientos, plan, manuales, normas y estándares
- Definición de nomenclatura y rotulación para gestión documentos:

2.2.1.2.2 Codificación

Se debe definir una codificación, para la identificación correcta de la información. Dicha codificación deberá ser usada por cada una de las empresas vinculadas al proyecto:

- [Contratista]-[Contrato] [Año]-[Dpto]-[Tipo de documento]-[Numeración]

2.2.1.2.3 Almacenamiento de documentación/información

Toda información generada será guardada físicamente en carpetas y la misma será debidamente identificada y rotulada con la codificación indicada. Paralelamente tal

información se pasará también a formato digital. Se creará un inventario que llevará el control de cada documento emitido o recibido.

2.2.1.2.4 Distribución de la información

Los documentos serán distribuidos de manera electrónica a través de la red interna de la compañía, FTP o vía correo si es que fuese necesario, los formatos utilizados para tal menester serán PDF, DOC, XLS...etc.

2.2.1.2.5 Identificación de documentos

La documentación debe tener los logos respectivos de cada sub-contratista.

2.2.2 SELECCIÓN DE METODOLOGÍA DE DESARROLLO

En el desarrollo del aplicativo Web que se desea implementar se ha elegido la metodología ágil denominada (extreme programming) XP, teniendo en cuenta los fundamentos teóricos investigados con antelación en el capítulo 1, no obstante se tomarán también algunos aspectos de la metodología de desarrollo SCRUM (orientada para proyectos que necesitan rapidez en cambios de requerimientos o requisitos en proyectos) debido a que esta metodología posee ciertos elementos que ayudan y complementan a XP. Se tomaron en referencia los siguientes puntos como premisas para la selección respectiva de la metodología:

- **Lapso de tiempo para el desarrollo:** Es relativamente corto y el mismo va acorde al avance del proyecto principal actual de TECNIE, el cual se encuentra en una etapa de finalización.
- **Cantidad y disponibilidad de recursos baja:** TECNIE ha manifestado no querer invertir en software, compra de equipos extra, ni más personal de trabajo (actualmente el equipo está conformado por un programador, un jefe de archivo técnico y tres usuarios de control de documentos) para la colaboración en el desarrollo del presente proyecto.

- **Complejidad del sistema a ser desarrollado:** La resolución de dificultades surgidas durante el desarrollo del proyecto debe ser simple y eficaz.
- **Vínculo relacional del cliente y el desarrollador:** A medida que el proyecto siga avanzando y surjan las necesidades el cliente siempre tendrá contacto con el desarrollador, esto le brinda al cliente una mayor comodidad debido a que el mismo va siendo partícipe de los avances y progresos del proyecto.

2.2.2.1 Planificación del proyecto usando la metodología elegida:

La planificación del proyecto se realizará de la siguiente manera:

- a) Se iniciará la planificación estableciendo el equipo de trabajo.
- b) Luego se definirán los requerimientos funcionales y no funcionales.
- c) Despues de esto se realizará el dimensionamiento del hardware.
- d) Se planteará el alcance que tendrá el sistema o aplicativo Web a ser desarrollado, para de esta manera empezar a determinar los profiles (roles) de usuario.
- e) Luego se definirán las historias y tareas de usuario, en donde se describirán brevemente los requerimientos del cliente para con ayuda del mismo analizar y establecer el tiempo estimado de desarrollo.
Todas estas historias y tareas de usuario serán clasificadas y ordenadas mediante conjuntos de iteraciones, que a su vez permitirán elaborar y actualizar el plan total de ejecución y desarrollo por etapas, delimitadas por cada iteración y deberán tener fechas de cumplimiento establecidas, con esto se podrá obtener una fecha tentativa de entrega del proyecto. Se debe tener en cuenta que siempre podrán surgir más historias y tareas de usuario durante el pleno desarrollo (las cuales se deben ir actualizando en el cuadro de planificación).
- f) Una vez establecido el plan se iniciará el diseño del prototipo del proyecto determinando los siguientes aspectos:
 - El diseño y elección de arquitectura de trabajo.

- El diseño de diagramas de clases y de bases de datos.
- La programación/codificación del código fuente.
- Pruebas unitarias durante el desarrollo, por cada historia de usuario.

g) Finalmente se realizarán las pruebas aceptación con el cliente(Usuario):

2.3 INICIALIZACIÓN DE DESARROLLO DEL PROYECTO

2.3.1 EQUIPO DE TRABAJO

El equipo de trabajo que facilitara el desarrollo y realización del presente proyecto está compuesto por cinco personas a continuación se las detalla indicando su posición y roles:

- Un **Desarrollador/Programador** del sistema gestor de información, quien es el cargado del desarrollo del aplicativo Web.
- Un **Coordinador Jefe de archivo técnico** (Jefe del área de control de documentación y encargado principal de captar/recopilar toda la información pertinente para preparar el Dossier de calidad/ingeniería/construcción). Se encarga de gestionar y facilitar recursos para las reuniones sobre avances del proyecto y de dirigir el proceso (hacer cumplir las fechas de entrega planteadas) de desarrollo, también brinda sugerencias durante el desarrollo y realiza pruebas de aceptación final.
- Tres **miembros de control de documentación** (uno en campo y dos en Quito). Dichos miembros se encargan de facilitar la información para el desarrollo de historias y tareas de usuario, realizar las pruebas de aceptación por historia de usuario del aplicativo Web a realizarse y de retroalimentar al desarrollador durante el desarrollo de cada historia de usuario.

2.3.2 REQUERIMIENTOS FUNCIONALES

El sistema gestor de información debe poder contar con los siguientes requerimientos funcionales:

2.3.2.1 Ingresar/Almacenar/Gestionar:

Información de Proyectos/Sub-Proyectos, Proveedores/Contratistas, Disciplinas, Alcances, Archivos, Equipos, Planos, Documentación técnica en general.

2.3.2.2 Generar/Proporcionar:

Consultas de documentación: técnica, contractual, de equipos, planos, manuales, catálogos, facturas, procedimientos, hojas de datos y otras en caso de requerirse.

2.3.3 REQUERIMIENTOS NO FUNCIONALES:

El sistema gestor de información debe poder contar con los siguientes requerimientos no funcionales:

- La Interfaz dinámica, simple y fácil de usar.
- Debe Administrar y asignar permisos para diferentes tipos de usuarios, clasificados en niveles determinados por TECNIE.
- Debe ser rápido y eficiente.
- Debe ser multi-usuario.
- La ejecución del sistema gestor de información podrá realizarse a nivel local (PC), interno (INTRANET) y externo (INTERNET).
- Personalización de consultas de acuerdo a los requerimientos del usuario surgidas durante el proceso de desarrollo del proyecto.
- El sistema debe contar con las seguridades respectivas y restricciones de acceso a usuarios no autorizados de modo que tenga integridad y confiabilidad de los datos.
- Respaldo periódico y automatizado de datos.

- El sistema debe posibilitar la verificación de un historial de edición de información.
- Debe tener un manual de gestión/edición de información del sistema.
- Debe garantizar que la información ingresada sea correcta y coherente

2.3.4 DIMENSIONAMIENTO DE HARDWARE

2.3.4.1 Cantidad de Información estimada por cada proyecto realizado

Después de la conversación sostenida con el jefe de control de documentación sobre los registros (cantidad y tipos) que se ingresarán en la base de datos y teniendo en cuenta los proyectos de ingeniería y fiscalización anteriores se estimaron los siguientes valores:

- Cantidad estimada de registros en la base de datos:3000 a 4500
(Se considera registros de logs).
- Cantidad estimada de archivos asociados a registros:1500
- Tamaño estimado por archivo (archivo principalmente PDFs):75MB – 150MB

Debido al tamaño excesivo por archivo se considera trabajar únicamente con enlaces o hipervínculos a archivos que estarán ubicados en un directorio FTP alojados en un servidor de la empresa o estación de trabajo, a los cuales se podrán acceder mediante autenticación de usuario y contraseña.

2.3.4.2 Cantidad de usuarios que utilizaran el sistema

Debido a la confidencialidad del proyecto, existe un número limitado de personas que tienen acceso al aplicativo Web, a continuación se presentan los usuarios:

- **Para consulta de datos:**
 - Usuarios internos (TECNIE): 5 personas, se considera un crecimiento del 40%. **Total=7.**

- Usuarios externo (Cliente-Objetivo): 4 personas, se considera un crecimiento del 50%. **Total=6.**

- **Para edición de datos:**

- Usuarios internos (TECNIE): 3 personas, se considera un crecimiento del 33%. **Total=4.**

- **Para administración del sistema:**

- Usuarios internos (TECNIE): 2 personas, se considera un crecimiento del 50%, **Total=3.**

- **Total:**

- Total de usuarios de consulta = 9 a 13 usuarios.
- Total de usuarios de edición = 3 a 4 usuarios.
- Total de usuarios de administración = 2 a 3 usuarios.
- **Total 14 a 20 usuarios.**

El Jefe de archivo técnico supo manifestar que ningún proyecto de fiscalización o ingeniería por más grande que sea incluirá una cantidad tan alta de usuarios. Del valor total adquirido 14 a 20 usuarios se toma como porcentaje de concurrencia de uso el 50%, lo que da como resultado 7 a 10 usuarios que se conectarán simultáneamente.

2.3.4.2.1 Análisis de requisitos de rendimiento

No es necesario detallar requisitos de rendimiento debido a que la cantidad de usuarios en concurrencia que usarán la base de datos no sobrepasan los 10 usuarios. Lo que implica que la carga de transacción es bastante baja y de esta manera no es requerida la utilización de un equipo de alto rendimiento. Por lo que únicamente es necesario que el equipo en donde se instale el software debe

soportar los requerimientos de las herramientas de desarrollo, servidores, capacidad de almacenamiento, procesamiento...etc.

Existen dos equipos brindados por la empresa TECNIE para realizar el desarrollo del software y para la presentación definitiva de la plataforma Web. En ninguno de los casos se invirtió en la compra (más adelante se detallaran sus características), puesto que pertenecen a la empresa.

2.3.5 ALCANCE DEL SOFTWARE

El sistema gestor de información S.G.I. (aplicativo Web) a ser realizado tiene como objetivo mejorar la gestión y organización de la información generada y obtenida durante el desarrollo de un proyecto de ingeniería/fiscalización realizado por TECNIE.

Este aplicativo Web permitirá gestionar y visualizar el almacenamiento de información final generada (planos, manuales, equipos, documentos técnicos, dossier de calidad/ingeniería/construcción...etc.) de un proyecto de ingeniería/fiscalización específico, por medio de una interfaz, la cual variará según el tipo de privilegio asociado al usuario.

Dicho sistema gestor de información será creado de tal manera que los miembros del área de control de documentos puedan gestionar y almacenar la información final generada de un proyecto en una base de datos (en texto plano) y enlazarla a documentos ubicados en una determinada locación (en un directorio específico: dicho directorio contendrá documentos digitales relacionados a planos, manuales, equipos, documentos técnicos, dossier de calidad, documentos legales, financieros...etc.).

Los miembros del área de control de documentación trabajarán conjuntamente con el(es) administrador(es) de red y/o sistemas, a los cuales solicitarán dado el caso que se requirido: la agregación de módulos adicionales, creación, edición y eliminación de usuarios, modificación de la interfaz gráfica (total o parcialmente),

ubicación de coordenadas para los planos que comprende la obra del proyecto, y cualquier cambio solicitado en función de los nuevos requerimientos que vayan surgiendo.

2.3.6 PERFILES DE USUARIOS DEL SISTEMA

Mediante conversaciones con el jefe del área de control de documentación se han definido los tipos y niveles de usuario para el sistema según las necesidades del entorno, acordándose lo siguiente:

Se crearan tres tipos de usuario: administrador, editor y lector, los cuales tendrán privilegios administración, edición y lectura. Todos estos usuarios deberán ser gestionados únicamente por el administrador(o un usuario administrador) del sistema gestor de información.

A continuación se detalla información referente a los niveles de usuario:

2.3.6.1 Usuarios internos

2.3.6.1.1 Administrador:

Usuario que posee todos los derechos para modificar el sistema, este puede asignar, editar, eliminar permisos y crear nuevos usuarios. Tiene también la capacidad de poder gestionar totalmente la información, es decir tiene acceso a todos los módulos del aplicativo Web,

Este usuario igualmente estará encargado de gestionar los contenidos a mostrarse en el sistema, realizar ingresos, actualizaciones y eliminaciones de cada módulo, el único que podrá tener acceso a los archivos fuente para edición de apariencia del sistema. Usuario Objetivo: Sys-Admin, Administrador de Red o Base de datos.

2.3.6.1.2 Editor:

Este usuario puede realizar la gestión de la información del sistema además de hacer consultas de información dentro del sistema, entre sus funciones están el

ingreso y edición de la misma. Usuario objetivo: Cualquier miembro del área de control de documentación, por petición del cliente igualmente está incluido el jefe de archivo de control de documentación.

2.3.6.1.3 *Lector:*

El usuario lector tiene acceso limitado solamente de consulta, pues solo podrá observar la información ingresada en el sistema, mas no podrá realizar ninguna edición. Usuario objetivo: Personal vinculado a la empresa (solo para consulta interna: Presidente, Gerente General, Gerente Financiero, Asesor de proyecto, Gerente técnico, Ingenieros de Planeación y control), y/o cliente objetivo específico externo (con el que esté trabajando la empresa) que esté relacionado directamente al proyecto en ejecución.

2.3.7 JUSTIFICACIÓN DE LA SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO

La elección de las herramientas se basó principalmente en los siguientes factores:

- **Compatibilidad** con el sistema operativo (Windows XP) del equipo destinado para el desarrollo y con el sistema operativo (Windows 8) del equipo de instalación definitiva para las pruebas de aceptación y entrega del aplicativo Web.
- **Universalidad** de herramientas más utilizadas en tecnología de desarrollo.
- **Flexibilidad, robustez y sencillez.**
- **Inversión en tecnología baja o nula**, mediante conversaciones con el jefe de archivo técnico se ha acordado en desarrollar e implementar el aplicativo Web con herramientas preferentemente que estén bajo licencia GPL(General Public License) para no generar costos adicionales.

Bajo tales premisas se han elegido las siguientes herramientas y tecnologías de desarrollo, además de otros criterios adicionales:

2.3.7.1 Servidor Web: Apache 2.2.8:

- Servidor Web más conocido en la actualidad.
- Flexible y robusto.
- Es abierto posee licencia GPL (General Public License).

2.3.7.2 Gestor de base de datos: MySQL 5.0.51b:

- Rápido y sencillo de programar.
- Se maneja con un lenguaje estructurado estandarizado de programación para bases de datos (SQL).
- Permite y facilita su manejo mediante una interfaz gráfica por medio de la instalación de una extensión: phpMyAdmin database manager 2.10.3.

2.3.7.3 Lenguaje de programación - Cliente: JavaScript:

- Permite generar una amplia interactividad con el usuario por medio de la creación de contenidos dinámicos.
- Trabaja conjuntamente con PHP y HTML sin problemas.

2.3.7.4 Lenguaje de programación – Servidor: PHP 5.2.6:

- Lenguaje orientado a objetos muy similar a C o C++, lo que facilita bastante la creación de código, por su simple sintaxis.
- Posee una muy buena velocidad de ejecución.
- Robusto y estable.
- Permite establecer seguridades(a distintos niveles).

A primera vista se según lo indicado se debería hacer uso de la plataforma de desarrollo Web WAMP (Windows, Apache, MySQL, PHP), sin embargo existe una plataforma de desarrollo Web adicional, de instalación mucho más sencilla y manejable que posee iguales potencialidades pero que resta ciertas configuraciones (que sin embargo no son preponderantes), se denomina:

- **AppServ Open Project con la versión 2.5.10.**

2.3.7.5 Entorno de desarrollo integrado: Netbeans 8.0.2 y Notepad++ 6.7.3

- Entre sus características más destacables están la versatilidad de trabajo que poseen (permiten trabajar con: HTML, PHP, CSS, JS, SQL), su facilidad de manejo y su peso liviano.
- Ambas aplicaciones permiten diseñar la aplicación Web (sistema gestor de información) y desarrollar las respectivas interfaces gráficas.

2.3.8 HISTORIAS DE USUARIO^{[41][42]}

A continuación se presentan las especificaciones y características solicitadas por los usuarios de la empresa (TECNIE) para el aplicativo Web. Tales especificaciones se caracterizan por ser relativamente cortas y simples, indicando concreta y sucintamente lo que va a realizar el sistema.

Cada historia de usuario contiene los siguientes elementos :

- **Número:** Identificador que posee cada historia de usuario.
- **Nombre historia:** Título que describe la historia de usuario.
- **Riesgo en desarrollo:** Es la complejidad que posee la historia de usuario para ser implementada, sus principales criterios de determinación son: tiempo estimado de desarrollo, complejidad de algoritmo a implementar y lenguajes de programación utilizados en el proceso.
- **Puntos estimados:** Indica el tiempo estimado de desarrollo, en el que se puede implementar la historia de usuario, usualmente se traduce en semanas.
- **Usuario:** Persona encargada de realizar las actividades descritas en la historia de usuario.

- **Prioridad en negocio:** Importancia y criticidad otorgada a la historia de usuario, puede ser alta, baja o media.
- **Descripción:** Descripción comprensible y resumida de la historia de usuario, que plasma los requerimientos funcionales especificados por el cliente.
- **Observaciones:** Son detalles adicionales para comprensión de los requerimientos del usuario, que permitirán implementar la historia del usuario.
- **Iteraciones:** Son las fases a seguir antes de que el sistema sea entregado. Cada iteración está compuesta por sus respectivas historias y tareas de usuario.
- **Tarea de usuario:** Son tareas creadas con el fin de dividir las historias de usuario para poder definir los tiempos estimados de entrega y obtener una mayor estimación del tiempo de duración de la historia de usuario. En un principio las tareas de usuario no tendrán fechas de inicio y fin, debido a que esto será implementado en el momento de estimación de esfuerzos.

2.3.8.1 Iteraciones^[43]

A continuación se realizan las historias y tareas de usuario según las prioridades analizadas con el cliente (se colocan un par de ejemplos, revisar Anexo).

2.3.8.1.1 Primera iteración

Historia de usuario	
Número:1	Nombre historia: Acceso al sistema
Usuario: Administrador, Editor, Lector	Prioridad en negocio: Alta
Puntos estimados: 3	Riesgo en desarrollo: Alto
Descripción: Los usuarios podrán ingresar al sistema mediante un nombre de usuario y una contraseña, que será otorgado por el administrador. El sistema deberá tener en su base de datos el tipo de usuario que se ha ingresado.	

Observaciones: Dependiendo del tipo de usuario que se ha ingresado, el sistema deberá cargar los módulos respectivos. Si algún individuo intenta ingresar y no posee credenciales, el sistema no permitirá el ingreso.

Tabla 2.1: Historia de usuario 1: Acceso al sistema

Tarea de usuario		
Número:T1	Historia: HU1	
Nombre: Acceder al sistema por medio login(Validación de datos de usuario)		
Tipo: Desarrollo	Fecha Inicio:	Fecha Fin:
Descripción: El usuario podrá acceder únicamente cuando su usuario y contraseña estén creados en el sistema.		

Tabla 2.2: Tarea de usuario 1: Acceder al sistema por medio login (Validación de datos de usuario)

Historia de usuario		
Número:2	Nombre historia: Gestión de contenedor: proyecto	
Usuario: Administrador, Editor		Prioridad en negocio: Alta
Puntos estimados: 1.4		Riesgo en desarrollo: Medio
Descripción: El administrador y editor del sistema se encargan de la edición de información de proyectos.		
Observaciones: Este contenedor de información es de tipo primario.		

Tabla 2.3: Historia de usuario 2: Gestión de contenedor: proyecto

Tarea de usuario		
Número:T2	Historia: HU2	
Nombre: Crear nuevo registro		
Tipo: Desarrollo	Fecha Inicio:	Fecha Fin:
Descripción: Se añade un nuevo registro proyecto en la base de datos del aplicativo Web.		

Tabla 2.4: Tarea de usuario 2: Crear nuevo registro

Tarea de usuario		
Número:T3	Historia: HU2	
Nombre: Listar registros existentes		
Tipo: Desarrollo	Fecha Inicio:	Fecha Fin:
Descripción: Se listan todos los proyectos registrados en la base de datos del aplicativo Web.		

Tabla 2.5: Tarea de usuario 3: Listar registros existentes

Tarea de usuario		
Número:T4	Historia: HU2	
Nombre: Modificar registro		

Tipo: Desarrollo	Fecha Inicio:	Fecha Fin:
Descripción: Se edita un proyecto registrado en la base de datos del aplicativo Web.		

Tabla 2.6: Tarea de usuario 4: Modificar registro

2.3.8.2 Estimación de tiempo y esfuerzos^[44]

Una vez definidas las historias y tareas de usuario con el cliente que van a ser desarrolladas e implementadas, es necesaria la estimación del tiempo y esfuerzo requerido para la compleción del proyecto, estableciendo las prioridades respectivas, de modo que se satisfaga al cliente a medida que el proyecto se encuentre en etapa de desarrollo.

Por lo que se hace totalmente indispensable organizar cada historia de usuario con sus respectivas tareas mediante iteraciones, para poder establecer plazos de tiempo de entrega.

2.3.8.2.1 Aspectos a considerarse:

La metodología XP hace hincapié en el desarrollo de código en varios equipos de 2 personas cada uno, sin embargo debido a las limitaciones de personal únicamente existirá un programador.

Por semana se considerarán 5 días laborables, de los cuales el tiempo diario invertido de desarrollo será de 4 horas. Se tendrán en cuenta las prioridades establecidas por historias de usuario, riesgo y complejidad de desarrollo, requerimientos y puntos estimados (el punto estimado se considerará como una semana).

Si bien es cierto se ha elaborado una planificación de tiempo para este proyecto esto no significa que los tiempos establecidos en el cronograma sean fijos debido a diversos factores, por lo que se considerara el aumento de un tiempo adicional del 40% más para efectos de:

- Análisis de requerimientos

- Reuniones de trabajo
- Cambios no previstos durante el desarrollo
- Pruebas unitarias por parte del programador
- Pruebas de aceptación por parte del cliente
- Otros imprevistos

A continuación en la Tabla 2.61 se representa de forma resumida la estimación de esfuerzos en cada historia de usuario incluyendo sus puntos estimados, prioridad y riesgo para el desarrollo y compleción del proyecto.

No.	Historia de usuario	Puntos E.	Prioridad	Riesgo
1	Acceso al Sistema	2.2	alta	Alto
2	Gestión de contenedor: Proyecto	1	alta	Medio
3	Gestión de contenedor: contratista	1	alta	Medio
4	Gestión de contenedor: disciplinas	1	alta	Medio
5	Gestión de contenedor: alcance	1	alta	Medio
6	Gestión de contenedor: contrato	2.2	alta	Alto
7	Gestión de contenedor: Sistema	1.4	alta	Alto
8	Gestión de contenedor: equipos	2.8	alta	Alto
9	Gestión de contenedor: archive	2.8	alta	Alto
10	Visualizar módulos de lectura de datos	2.2	media	Bajo
11	Mostrar datos de login en el sistema	0.3	media	Bajo
12	Modificar datos usuarios del sistema	1.4	alto	Medio
13	Mostrar lista de usuarios del sistema	0.8	media	Bajo
14	Visualizar registro de edición “Logs”	1.4	alta	Alto
15	Publicar contenido y actualizar en el sistema.	2.2	alta	Bajo
16	Gestión de Contenedores y Mantenimiento del sistema	2.8	media	Alta

Tabla 2.61: Estimación de esfuerzos

En la Tabla 2.62 se muestra la equivalencia en horas de los puntos estimados utilizados para la compleción de cada historia de usuario y a los cuales se aumentan el porcentaje extra por los motivos ya indicados con anterioridad. De esta manera se establecen los lapsos de tiempos de entrega de definitivos.

No.	Historia de usuario	Puntos Estimados.	Horas	Puntos Estimados. 40%	Horas	adi.40% Horas
1	Acceso al sistema	2.2	44	3	60	16
2	Gestión de contenedor: proyecto	1	20	1,4	28	8
3	Gestión de contenedor: contratista	1	20	1,4	28	8
4	Gestión de contenedor: disciplinas	1	20	1,4	28	8
5	Gestión de contenedor: alcance	1	20	1,4	28	8
6	Gestión de contenedor: contrato	2.2	44	3	60	16
7	Gestión de contenedor: sistema	1.4	28	2	40	12
8	Gestión de contenedor: equipos	2.8	56	4	80	24
9	Gestión de contenedor: archivo	2.8	56	4	80	24
10	Visualizar módulos de lectura de datos	2.2	44	3	60	16
11	Mostrar datos de login en el sistema	0.3	7	0,4	8	1
12	Modificar datos usuarios del sistema	1.4	28	2	40	12
13	Mostrar lista de usuarios del sistema	0.8	16	1	20	4
14	Visualizar registro de edición “Logs”	1.4	28	2	40	12
15	Publicar contenido y actualizar en el sistema.	2.2	44	3	60	16
16	Gestión de Contenedores y Mantenimiento del sistema	2.8	56	4	80	24
TOTAL		26.5	531	37	740	209

Tabla 2.62: Estimación de esfuerzos con tiempos adicionales

A continuación se agrupan y clasifican las historias de usuario, esto se realiza para definir las entregas formales de las iteraciones en este proyecto se determinaron cuatro iteraciones, tal y como se muestra en la Tabla 2.63.

No.	Historia de usuario	Puntos E.	Prioridad	Riesgo	Iteración
1	Acceso al sistema	3	alta	alto	1
2	Gestión de contenedor: proyecto	1,4	alta	medio	1
3	Gestión de contenedor: contratista	1,4	alta	medio	1
4	Gestión de contenedor: disciplinas	1,4	alta	medio	1
5	Gestión de contenedor: alcance	1,4	alta	medio	1
6	Gestión de contenedor: contrato	3	alta	alto	2
7	Gestión de contenedor: sistema	2	alta	alto	2
8	Gestión de contenedor: equipos	4	alta	alto	2
9	Gestión de contenedor: archivo	4	alta	alto	2

10	Visualizar módulos de lectura de datos	3	media	bajo	3
11	Mostrar datos de login en el sistema	0,4	media	bajo	3
12	Modificar datos usuarios del sistema	2	alto	medio	3
13	Mostrar lista de usuarios del sistema	1	media	bajo	3
14	Visualizar registro de edición "Logs"	2	alta	alto	3
15	Publicar contenido y actualizar en el sistema.	3	alta	bajo	4
16	Gestión de Contenedores y Mantenimiento del sistema	4	media	alta	4

Tabla 2.63: Historias de usuario con iteraciones

2.3.8.3 Planificación de entregas

2.3.8.3.1 Planificación de entregas de historias de usuario

Mediante la ayuda de los puntos estimados obtenidos para cada historia de usuario y consecuentemente de cada iteración, se procede a establecer las fechas de inicio y finalización de cada historia de usuario, hay que considerar que dentro del plan de entregas previsto se procede a entregar al usuario los parciales de cada iteración, por lo que en total se van a realizar cuatro entregas, se considera la fecha de finalización de cada iteración como fecha de entrega (esta subrayado en color verde claro), tal y como se muestra en la siguiente tabla 2.64.

Historia de usuario	Comienzo	Fin	Iteración
Acceso al sistema	23/09/2013	11/10/2013	1
Gestión de contenedor: proyecto	14/10/2013	22/10/2013	1
Gestión de contenedor: contratista	23/10/2013	31/10/2013	1
Gestión de contenedor: disciplinas	01/11/2013	11/11/2013	1
Gestión de contenedor: alcance	12/11/2013	20/11/2013	1
Gestión de contenedor: contrato	21/11/2013	11/12/2013	2
Gestión de contenedor: sistema	12/12/2013	25/12/2013	2
Gestión de contenedor: equipos	26/12/2013	22/01/2014	2
Gestión de contenedor: archivo	23/01/2014	19/02/2014	2

Visualizar módulos de lectura de datos	20/02/2014	12/03/2014	3
Mostrar datos de login en el sistema	13/03/2014	26/03/2014	3
Modificar datos usuarios del sistema	27/03/2014	09/04/2014	3
Mostrar lista de usuarios del sistema	10/04/2014	16/04/2014	3
Visualizar registro de edición “Logs”	17/04/2014	07/05/2014	3
Publicar contenido y actualizar en el sistema.	08/05/2014	28/05/2014	4
Gestión de Contenedores y Mantenimiento del sistema	29/05/2014	25/06/2014	4

Tabla 2.64: Fechas de planificación de entrega de historias de usuario

2.3.8.3.2 Planificación de entregas de Historias de usuario y Tareas de usuario:

De la misma manera en la Tabla 2.65 con ayuda de los puntos estimados se calculan y definen los tiempos de desarrollo de cada una de las tareas de usuario pertenecientes a cada historia correspondiente, como se muestra a continuación.

Iteración	HU	TU	tarea de usuario	Puntos	Fecha I.	Fecha F.
1	1	1	Acceder al sistema por medio login	3	23/09/2013	11/10/2013
1	2	2	Crear nuevo registro Listar registros existentes Modificar registro Eliminar registro	1,4	14/10/2013	15/10/2013
1	2	3			16/10/2013	17/10/2013
1	2	4			18/10/2013	21/10/2013
1	2	5			22/10/2013	22/10/2013
1	3	6	Crear nuevo registro Listar registros existentes Modificar registro Eliminar registro	1,4	23/10/2013	24/10/2013
1	3	7			25/10/2013	28/10/2013
1	3	8			29/10/2013	30/10/2013
1	3	9			31/10/2013	31/10/2013
1	4	10	Crear nuevo registro Listar registros existentes Modificar registro Eliminar registro	1,4	01/11/2013	04/11/2013
1	4	11			05/11/2013	06/11/2013
1	4	12			07/11/2013	08/11/2013
1	4	13			11/11/2013	11/11/2013
1	5	14	Crear nuevo registro Listar registros existentes Modificar registro Eliminar registro	1,4	12/11/2013	13/11/2013
1	5	15			14/11/2013	15/11/2013
1	5	16			18/11/2013	19/11/2013
1	5	17			20/11/2013	20/11/2013
2	6	18	Crear nuevo registro Listar registros existentes Modificar registro	3	21/11/2013	26/11/2013
2	6	19			27/11/2013	02/12/2013
2	6	20			03/12/2013	06/12/2013

2	6	21	Eliminar registro		09/12/2013	11/12/2013	
2	7	22	Crear nuevo registro Listar registros existentes Modificar registro Eliminar registro	2	12/12/2013	16/12/2013	
2	7	23			17/12/2013	18/12/2013	
2	7	24			19/12/2013	23/12/2013	
2	7	25			24/12/2013	25/12/2013	
2	8	26	Crear nuevo registro	4	26/12/2013	01/01/2014	
2	8	27	Listar registros existentes Modificar registro Eliminar registro		02/01/2014	08/01/2014	
2	8	28			09/01/2014	15/01/2014	
2	8	29			16/01/2014	22/01/2014	
2	9	30	Crear nuevo registro Listar registros existentes Modificar registro Eliminar registro	4	23/01/2014	29/01/2014	
2	9	31			30/01/2014	05/02/2014	
2	9	32			06/02/2014	12/02/2014	
2	9	33			13/02/2014	19/02/2014	
3	10	34	Creación de módulos de consulta	3	20/02/2014	12/03/2014	
3	11	35	Visualizar datos de usuario autenticado	0,4	13/03/2014	26/03/2014	
3	12	36	Crear usuario del sistema Eliminar usuarios del sistema Modificar tipo de usuario Modificar datos del usuario autenticado	2	27/03/2014	31/03/2014	
3	12	37			01/04/2014	02/04/2014	
3	12	38			03/04/2014	07/04/2014	
3	12	39			08/04/2014	09/04/2014	
3	13	40	Listar registros existentes	1	10/04/2014	16/04/2014	
3	14	41	Listar registros existentes de edición	2	17/04/2014	07/05/2014	
4	15	42	Modificar y Actualizar contenido	3	08/05/2014	28/05/2014	
4	16	43	Modificar la base de datos. Realizar mantenimiento del sistema	4	29/05/2014	11/06/2014	
4	16	44			12/06/2014	25/06/2014	

Tabla 2.65: Fechas de planificación de entrega de historias y tareas de usuario

2.3.8.3.3 Análisis y tiempos totales de entrega por iteraciones

Según se puede notar en la tabla 2.66, en la **primera entrega** se presentará la autenticación de usuario y la administración de los módulos principales: proyecto, contratista, disciplina y alcance lo cuales servirán como soporte base para la creación de registros en los módulos secundarios, el tiempo de entrega de esta es de aproximadamente dos meses.

En la **segunda entrega** y más larga de todas se hará énfasis la administración de los módulos secundarios: archivo, equipos, proyecto y sistemas, la misma posee un tiempo mayor de entrega debido a la complejidad de desarrollo ya que en esta

segunda entrega es en donde se logra concretar el ingreso objetivo de registros finales que son los que serán mostrados en los módulos de visualización para la consulta.

Respecto a la **tercera entrega** se puede verificar la agregación de los diferentes módulos de consulta, la visualización de datos del usuario que inicio sesión, el módulo de administración de usuarios y el módulo de verificación de edición de registros con un tiempo de realización similar al de la primera entrega, por lo que se considera que en esta entrega ya están realizadas la mayoría de historias que conforman el aplicativo Web, debido a que las ultimas historias pertenecientes a la **cuarta iteración** solo consideran la realización de edición de contenido y bases de datos, no obstante por cualquier eventualidad o requerimiento suscitado durante el desarrollo que haya quedado pendiente se lo realizará en esta etapa final. A continuación se muestra el tiempo estimado de entrega por iteración en la Tabla 2.66.

Iteración	Entrega	Horas	Semanas	Meses
1ra	1ra	172	8.6	2.15
2da	2da	260	13	3.25
3ra	3ra	168	8.4	2.1
4ta	4ta	140	7	1.75
TOTAL		740	37	9.25

Tabla 2.66 Tiempos totales de entrega por iteraciones

2.4 DISEÑO DEL PROTOTIPO DE SOFTWARE

2.4.1 ELECCIÓN DE LA ARQUITECTURA

La arquitectura de software elegida para el proyecto es la de cliente-servidor de dos capas. En donde el cliente hace peticiones al servidor y este le brinda una respuesta al mismo, tal y como se muestra en la Figura 2.3.

Existen tres posibilidades de conexión entre el cliente y el servidor, a) pueden residir en el mismo computador b) en computadores diferentes dentro de una red local o intranet o c) el cliente y el servidor se pueden comunicar a través del internet.

De acuerdo a los requerimientos de la empresa TECNIE, se implementarán todas las posibilidades.

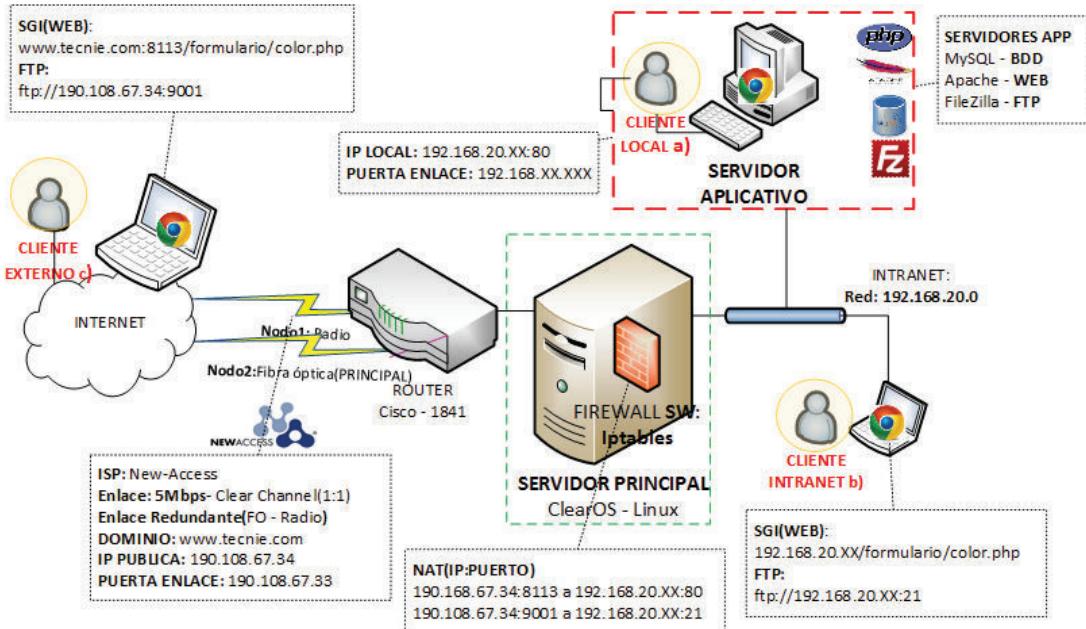


Figura 2.3: Arquitectura cliente-servidor

2.4.2 DIAGRAMA DE CLASES

Según las directrices de la metodología XP, el diagrama de clases presentado es el resultado de una consolidación del desarrollo de cada iteración y sirve como guía para la programación del aplicativo Web mas no es algo rígido ya que durante y después de la ejecución del proyecto se pueden aumentar o cambiar las clases dependiendo de las necesidades del cliente objetivo.

El diagrama está compuesto por las siguientes clases:

- Login, validación, permisosusuarios, seguridad, conexión
- Usuarios ,Administrador, lector, editor
- Registros, Ingresodatos, contrato
- Consulta, Mapa, sistemasyequipos, Logs

A continuación se presenta el diagrama de clases del proyecto SGI, dicho desarrollado con ayuda del software StarUML, representado en la Figura 2.4.

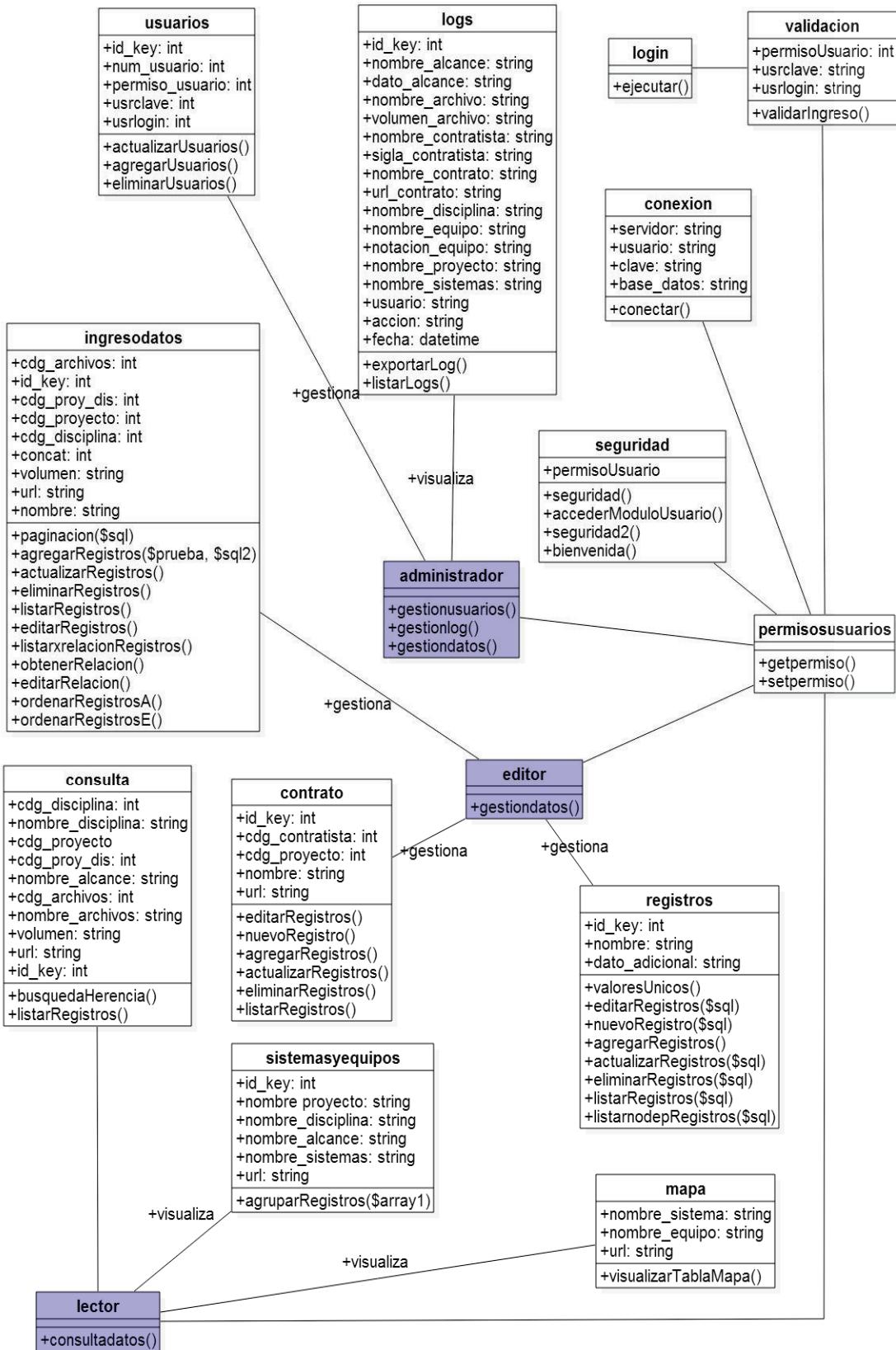


Figura 2.4: diagrama de clases de SGI

2.4.3 DIAGRAMA DE BASES DE DATOS

La Figura 2.5 muestra el diagrama relacional de base de datos, desde dicha base se almacenan todos los registros que los editores y administrador(es) han guardado, así mismo todas las consultas de información se realizan haciendo peticiones a esta base de datos, su elaboración es realizada en la herramienta MySQL Workbench.

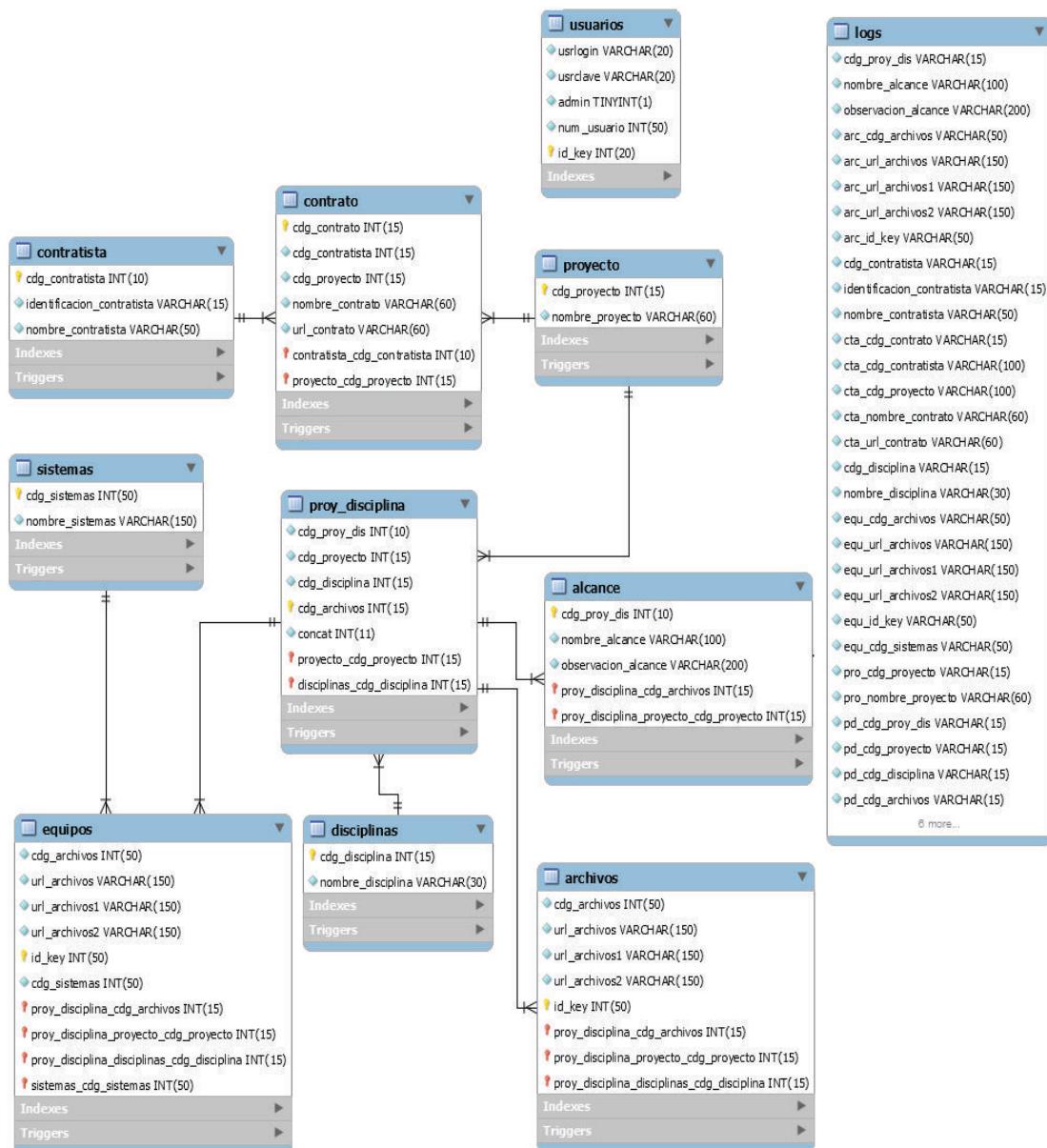


Figura 2.5: diagrama relacional de base de datos de SGI

2.4.3.1 Diccionario de datos

El diccionario de datos de las entidades/tablas respectivas utilizadas en el diagrama de bases de datos contiene los siguientes atributos y sus características.

- **Entidad:** Es el nombre de la entidad/tabla.
- **Detalle:** Es la descripción de los datos que se almacenaran en la entidad.
- **Atributos:** Es el nombre con el que se identifica a los atributos o campos.
- **Descripción:** Es el detalle otorgado al valor que será almacenado

A continuación se presenta el diccionario de datos de las entidades más importantes de la base de datos, comprendido por cada una de las tablas presentes en la misma.

Entidad: alcance.

Detalle: Almacena los alcances pertenecientes a cada disciplina.

Atributos	Tipo de dato	Clave Primaria	Clave Foránea	Auto-Incre.	Valor Nulo	Descripción
<u>cdg_proy_dis</u>	integer	x		x		clave primaria de alcance
<u>nombre_alcance</u>	varchar(150)					nombre de alcance

Tabla 2.67: diccionario de datos de la entidad alcance

Entidad: archivos.

Detalle: Almacena los registros/archivos pertenecientes a cada alcance.

Atributos	Tipo de dato	Clave Primaria	Clave Foránea	Auto-Incre.	Valor Nulo	Descripción
<u>cdg_archivos</u>	integer		x			clave foránea de la entidad proy_disciplina
<u>url_archivos</u>	varchar(150)					Especificación de archivo

url_archivos1	varchar(150)					Volumen
url_archivos2	varchar(150)					Url_archivo
id_key	integer	x				clave primaria de archivos

Tabla 2.68: diccionario de datos de la entidad archivos

Entidad: contratista.

Detalle: Almacena los registros/contratistas pertenecientes al proyecto.

Atributos	Tipo de dato	Clave Primaria	Clave Foránea	Auto-Incre.	Valor Nulo	Descripción
cdg_contratista	integer	x		x		clave primaria de contratista
identificacion_contratista	varchar(50)					siglas
nombre_contratista	varchar(150)					nombre de contratista

Tabla 2.69: diccionario de datos de la entidad contratista

Entidad: contrato.

Detalle: Almacena los registros/contratos pertenecientes a cada alcance.

Atributos	Tipo de dato	Clave Primaria	Clave Foránea	Auto-Incre.	Valor Nulo	Descripción
cdg_contrato	integer	x		x		clave primaria de contrato
cdg_contratista	integer		x			clave foránea de la entidad contratista
cdg_proyecto	integer		x			clave foránea de la entidad proyecto
nombre_contrato	varchar(150)					nombre de contrato
url_contrato	varchar(100)					url_contrato

Tabla 2.70: diccionario de datos de la entidad contrato

Entidad: disciplinas:

Detalle: Almacena los registros/disciplinas pertenecientes a cada proyecto.

Atributos	Tipo de dato	Clave Primaria	Clave Foránea	Auto-Incre.	Valor Nulo	Descripción
<u>cdg_disciplina</u>	integer	x		x		clave primaria de disciplina
<u>nombre_disciplina</u>	varchar(100)					nombre de disciplina

Tabla 2.71: diccionario de datos de la entidad disciplinas

Entidad: equipos.

Detalle: Almacena los registros/equipos pertenecientes a cada sistema.

Atributos	Tipo de dato	Clave Primaria	Clave Foránea	Auto-Incre.	Valor Nulo	Descripción
<u>cdg_archivos</u>	integer		x			clave foránea de la entidad proy_disciplina
<u>url_archivos</u>	varchar(150)					Nombre de equipo
<u>url_archivos1</u>	varchar(150)					notación
<u>url_archivos2</u>	varchar(150)					url_archivo
<u>id_key</u>	integer	x				clave primaria de equipos
<u>cdg_sistemas</u>	integer		x			clave foránea de la entidad sistemas

Tabla 2.72: diccionario de datos de la entidad equipos

Entidad: proyecto.

Detalle: Almacena los registros/sub-proyectos pertenecientes al proyecto principal.

Atributos	Tipo de dato	Clave Primaria	Clave Foránea	Auto-Incre.	Valor Nulo	Descripción
<u>cdg_proyecto</u>	integer	x		x		clave primaria de proyecto
<u>nombre_proyecto</u>	varchar(100)					nombre de proyecto

Tabla 2.73: diccionario de datos de la entidad proyecto

Entidad: proy_disciplina.

Detalle: Almacena los registros/vínculos pertenecientes a proyecto-disciplina-alcance-archivo.

Atributos	Tipo de dato	Clave Primaria	Clave Foránea	Auto-Incre.	Valor Nulo	Descripción
cdg_proy_dis	integer		x			clave foránea de la entidad alcance
cdg_proyecto	integer		x			clave foránea de la entidad proyecto
cdg_disciplina	integer		x			clave foránea de la entidad disciplina
cdg_archivos	integer	x		x		clave primaria de proy_disciplina
concat	integer		x			clave foránea de equipos/archivos

Tabla 2.74: diccionario de datos de la entidad proy_disciplina

Entidad: sistemas.

Detalle: Almacena los registros/sistemas pertenecientes al alcance.

Atributos	Tipo de dato	Clave Primaria	Clave Foránea	Auto-Incre.	Valor Nulo	Descripción
cdg_sistemas	integer	x				clave primaria de sistemas
nombre_sistemas	varchar(150)					nombre de sistemas

Tabla 2.75: diccionario de datos de la entidad sistemas

Entidad: usuarios.

Detalle: Almacena los registros/usuarios.

Atributos	Tipo de dato	Clave Primaria	Clave Foránea	Auto-Incre.	Valor Nulo	Descripción
usrlogin	varchar(20)					nombre de usuario

usrclave	varchar(20)					clave de usuario
admin	tinyint(1)					identificador tipo de usuario
num_usuario	integer					numeración
id_key	integer	x		x		clave primaria de usuarios

Tabla 2.76: diccionario de datos de la entidad usuarios

Entidad: usuario_activo.

Detalle: Almacena el registros/usuario activo (nombre de usuario en logs) - dato variable.

Atributos	Tipo de dato	Clave Primaria	Clave Foránea	Auto-Incre.	Valor Nulo	Descripción
id_usuarioa	integer	x				clave primaria de usuario activo
nombre_usuarioa	varchar(20)					nombre de usuario activo

Tabla 2.77: diccionario de datos de la entidad usuario_activo

2.4.4 DISEÑO DE INTERFAZ GRÁFICA DEL USUARIO

2.4.4.1 Spikes

El diseño de la interfaz gráfica se lo realizará con ayuda de “spikes”. Los spikes son bosquejos creados por el cliente y el desarrollador sobre el aspecto visual tentativo que deberá tener el aplicativo Web y que permitirán obtener un modelo objetivo de la interfaz gráfica final que se desarrollará.

Se debe tener en cuenta que dichos bosquejos podrán variar a lo largo del desarrollo y también estos serán usados como plantillas repetitivas en donde se definirán los componentes y controles de cada módulo, en algunos casos solo variarán datos como colores, texto, imágenes...etc.

Las plantillas se implementarán mediante la creación de hojas de estilo (CSS), mismas que serán realizadas acorde al requerimiento planteado del modelo de la

interfaz bosquejada. En las plantillas igualmente se podrán conservar aspectos como colores, textos, formato...etc.

2.4.4.2 Características generales de una interfaz gráfica

- Debe mantener en lo posible la misma distribución, estructura y colores para que sea de fácil aprendizaje.
- Debe ser de un fácil entendimiento tanto en su funcionamiento y mecánica.
- Debe brindar información efectiva y ordenada.

2.4.4.3 Características principales de la interfaz gráfica de SGI

Entre las principales características solicitadas por el cliente están las siguientes:

- En las plantillas principales se definirá el logo de la empresa, el nombre de la aplicación y datos adicionales (texto, imagen...etc.) que desee el usuario.
- Acceso a todos los módulos de visualización, vistas y listas desplegables de información previamente ingresada.
- La interfaz gráfica deberá ser compuesta por 2 o 3 plantillas principales, dependiendo y variando del módulo al que vaya a acceder.

A continuación se definen las características mostradas a cada tipo de usuario:

2.4.4.3.1 Características del usuario lector

El usuario lector posee todas las características principales mencionadas puede visualizar contenido de información a los módulos que disponga acceso.

2.4.4.3.2 Características del usuario editor

Este usuario posee las características del usuario lector y tiene adicionalmente la capacidad de editar contenido, por lo que deberá tener un acceso al enlace denominado “Edicion”, al cual tendrá la posibilidad de ingresar para editar la información que se visualiza en el sistema principal en cada módulo. Se mostrarán menús de edición, controles de ingreso y edición de registros y botones de navegación.

2.4.4.3.3 Características del usuario administrador

Este usuario tiene mayor control ya que posee la capacidad realizar acciones de un usuario editor y adicionalmente tiene la capacidad de administrar los usuarios del sistema y verificación de logs (edición de registros). Este deberá tener la capacidad de acceder a un enlace denominado “ADMINISTRACION” en donde tendrá acceso a menús de edición: controles de ingreso, edición de usuarios, botones de navegación y de consulta de logs.

A continuación se presentan los principales spikes o bosquejos que se desarrolló en conjunto con el jefe de control de documentos y el desarrollador del aplicativo Web en el programa Microsoft Office Power Point, los cuales al final se irán adaptando mediante hojas de estilo (CSS):

Interfaz inicial del sistema: El spike mostrado en la Figura 2.5 muestra la interfaz que se verá inicialmente al momento del ingreso al sistema.

Interfaz de acceso al sistema: El spike mostrado en la Figura 2.7 permite visualizar la interfaz de autenticación del usuario, en donde se colocaran el usuario y contraseña para el ingreso al sistema.

Interfaz del menú principal del sistema: En este spike se muestra el menú principal luego de haber accedido correctamente al sistema, tal y como se muestra en la Figura 2.8.



Figura 2.6: spike de interfaz inicial



Figura 2.7: spike de interfaz acceso



Figura 2.8: spike de menú principal

Interfaces de vistas de consulta del aplicativo Web: En los siguientes Spikes mostrados a continuación en las Figuras 2.9, 2.10 y 2.11 se muestran algunos de los módulos de vista al cual puede acceder todo usuario en el sistema.



Figura 2.9: spike de interfaz de una vista de consulta 1



Figura 2.10: spike de interfaz de una vista de consulta 2

Interfaz de edición de registros del aplicativo Web: El Spike mostrado en la Figura 2.12 representa el módulo de edición.

Interfaz de módulo primario de edición de registros del sistema: El Spike mostrado en la Figura 2.13 representa el módulo de edición “Proyecto”.

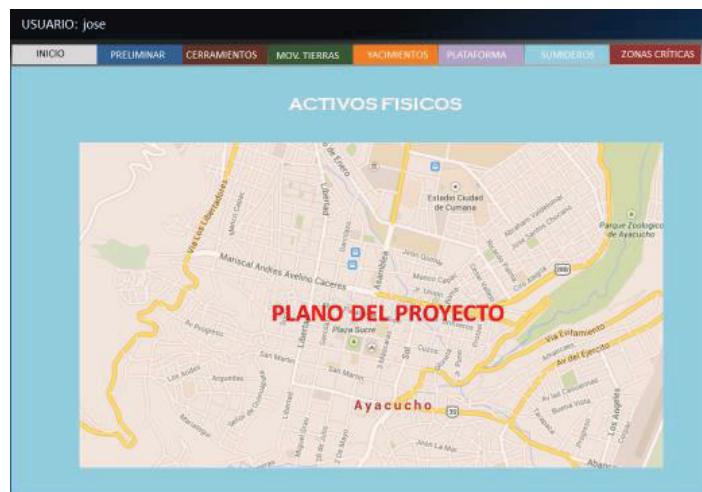


Figura 2.11: spike de interfaz de una vista de consulta 3



Figura 2.12: spike de módulo de edición



Figura 2.13: spike de interfaz de módulo primario (Proyecto) de edición de registros

Interfaz de módulo secundario de edición de registros del sistema: El Spike mostrado en la Figura 2.14 representa el módulo de edición “Archivos”.



Figura 2.14: spike de interfaz de módulo secundario (Archivos) de edición de registros

Interfaz de edición de usuarios del sistema: El Spike mostrado en la Figura 2.15 pertenece al menú presentado al administrador del sistema, el cual le permite editar usuarios.

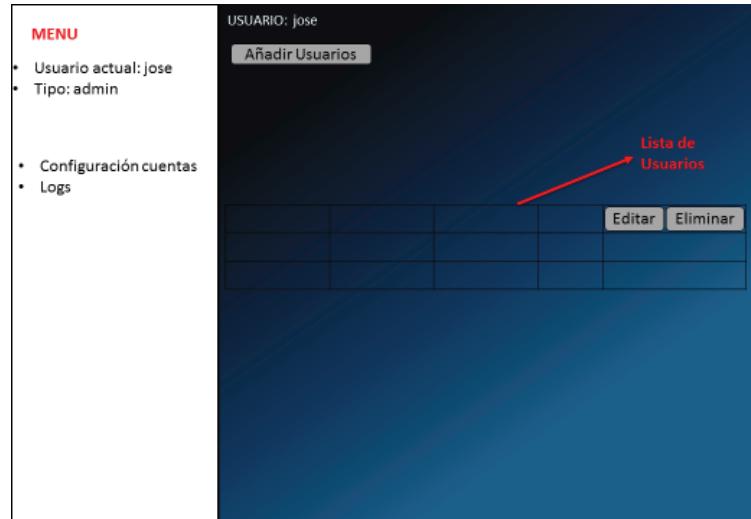


Figura 2.35: spike de interfaz de edición de usuarios

Interfaz de módulo de logs: El Spike mostrado en la Figura 2.16 representa al menú de administrador del sistema, que permite revisar los logs(registros de ediciones realizadas en el aplicativo Web).

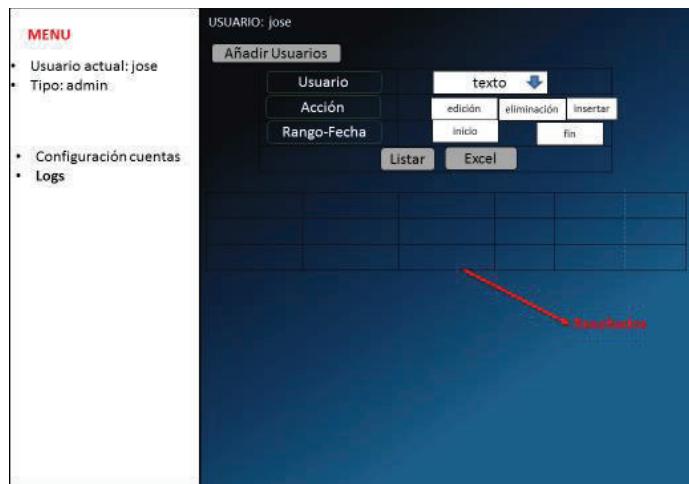


Figura 2.46: spike de interfaz de módulo de logs

2.4.4.4 Diseño e implementación de interfaz gráfica usando hojas de estilo

2.4.4.4.1 Cabecera

A continuación se definen cada uno de los elementos de la cabecera:

- **Título del proyecto:** Indica el título de todo el proyecto, el mismo puede ser modificado (recuadro amarillo).
- **Usuario:** Indica el nombre del usuario que inició una sesión.
- **Edición:** Enlace que lleva a una nueva ventana en donde se podrá ingresar y editar información relativa a todo el proyecto (dicho enlace solo está visible para usuarios con permisos de edición y administradores).
- **Administración:** Enlace visible únicamente para usuarios de tipo administrador, dicho enlace llevará a una nueva ventana en donde se podrán realizar las siguientes acciones: crear nuevos usuarios, modificar permisos, revisar logs de todo el sistema.

- **Salir:** Enlace permitirá al usuario actual (que se encuentre autenticado) salir de su sesión.

En la Figura 2.17 se representa a la cabecera principal.



Figura 2.57: cabecera de plantilla

2.4.4.4.2 Pantalla inicial

Al cargar la página del índice (página inicial) para ingreso al aplicativo Web aparecerá la pantalla para acceder al sistema (se debe tener en cuenta que los nombres e imágenes presentados son editables). Se muestra el botón “INICIO” en el cual se debe dar clic para acceder a la siguiente pantalla mostrada en la Figura 2.18.



Figura 2.68: pantalla inicial

2.4.4.3 Ingreso de credenciales

Aparece la siguiente página representada por medio de la Figura 2.19 que solicita una autenticación del usuario.

Se ingresan las credenciales y se da clic en el botón “INICIAR SESIÓN”.



Figura 2.19: ingreso de credenciales

Existen tres tipos de usuario: Administrador, Editor y Lector, los cuales son configurados/editados únicamente por el administrador del aplicativo Web.

A continuación se detalla información referente a los tipos de usuario:

- **Administrador:** Es un usuario que posee todos los derechos para modificar el sistema, el mismo puede asignar, delegar, editar permisos y crear nuevos usuarios. A continuación se presenta la interfaz y opciones a las que tiene acceso este usuario, en la Figura 2.20.
- **Lector:** El usuario lector tiene acceso a un servicio limitado de consulta, pues solo podrá observar la información ingresada en el sistema, mas no podrá realizar ninguna edición. Se puede constatar claramente en la Figura 2.22 que el usuario lector no posee permisos de edición.
- **Editor:** Este usuario puede únicamente realizar el ingreso y edición de información en la base de datos del sistema, la interfaz del usuario editor se representa en la Figura 2.21.



Figura 2.20: cabecera de administrador



Figura 2.21: cabecera de editor



Figura 2.22: cabecera de lector

2.4.4.4 Menú principal

Inmediatamente se accede al menú principal mostrado en la Figura 2.23, aquí se presentan dos maneras de acceso a los módulos de visualización de datos

ingresados en el sistema, la primera es ingresando a través de los hipervínculos colocados en la mitad de cada empresa de contrato (amarillo) y la segunda ingreso a través de las pestañas colocadas a lo largo de la barra (rojo). Ambas maneras direccionan a los usuarios a los módulos de visualización.

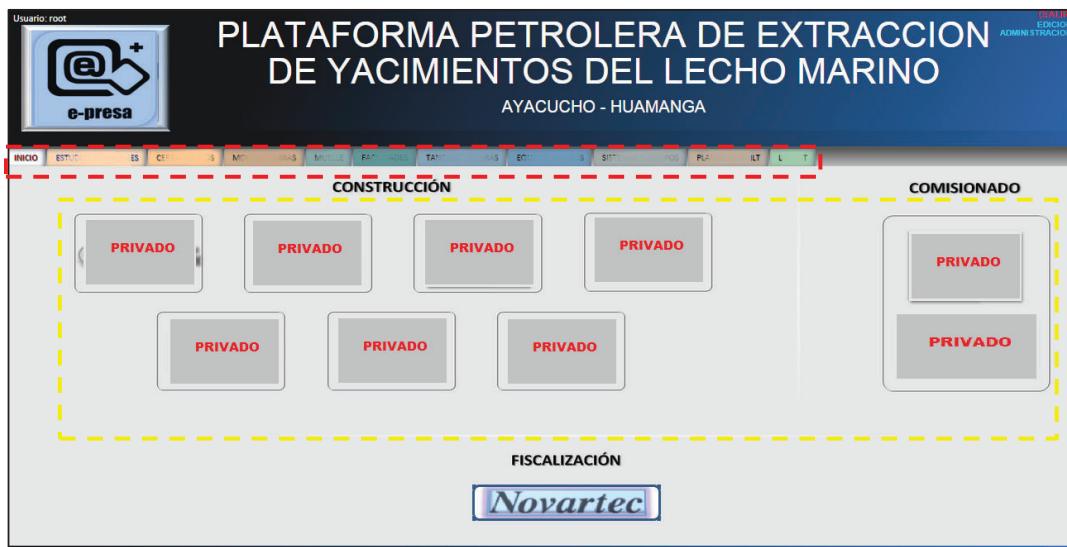


Figura 2.23: menú principal

2.4.4.4.5 Vista de consulta

Ahora accedemos a cualquiera de los módulos de visualización usando cualquiera de las dos maneras mencionadas, se presentará la siguiente pantalla mostrada en la Figura 2.24.

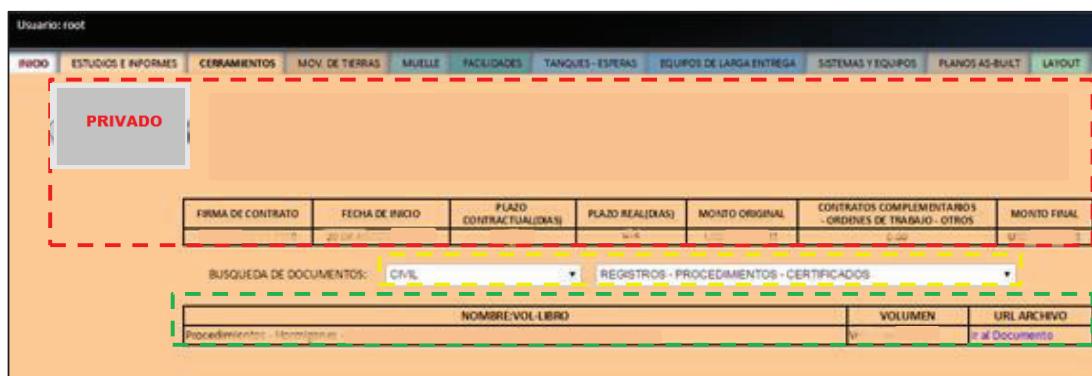


Figura 2.24: vista de consulta 1

Donde se divisarán la imagen e información perteneciente a la empresa colaboradora relacionada al proyecto (rojo), igualmente se presentarán los las listas de selección o combo-box (amarillo) respectivos que permitirán el acceso y búsqueda de la información acorde a los parámetros establecidos (en este caso “Disciplina” y “Alcance”).

La información consultada está encerrada en color verde. Al final del cuadro está un enlace “Ir al documento” con color azul, este enlace es el hipervínculo para llegar al documento objetivo de la consulta.

Siguiendo con la visualización de consultas ahora nos dirigimos al módulo de “SISTEMAS Y EQUIPOS” que se visualiza en la Figura 2.25, en donde encontraremos la información almacenada de los equipos en el proyecto ordenados mediante sistemas, dentro de cada registro de archivo existen dos enlaces un hipervínculo para acceder al archivo digital objetivo del equipo y un enlace para acceder a la ubicación del equipo en un mapa, tal como se observa a continuación.

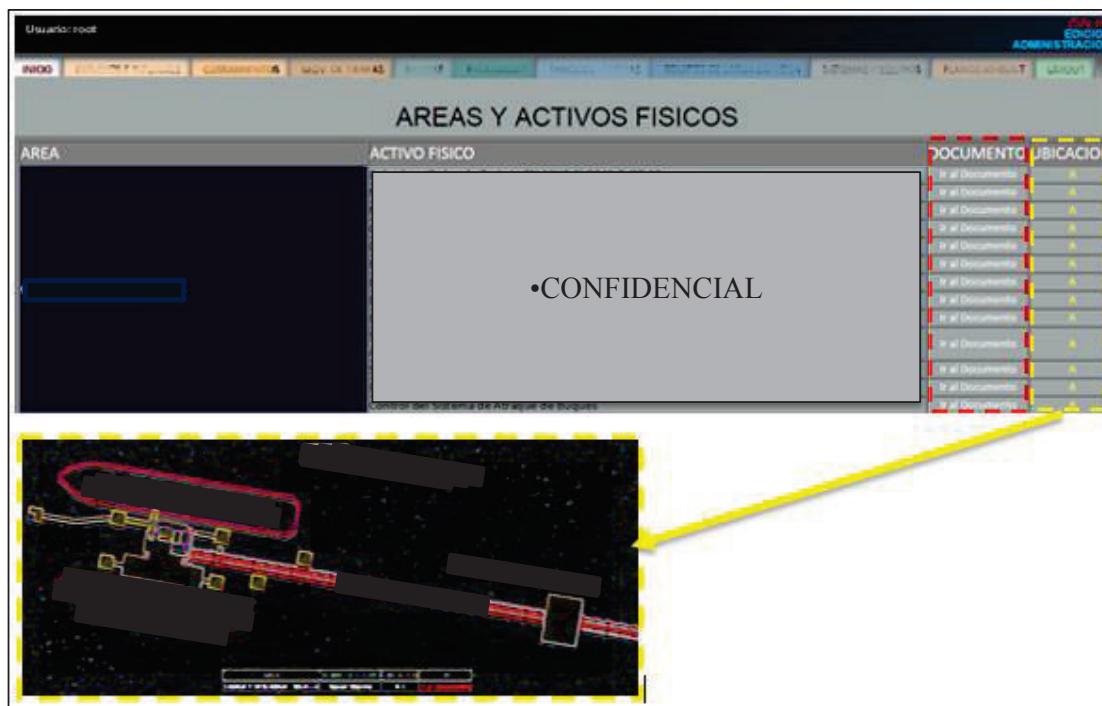


Figura 2.25: vista de consulta 2

El módulo LAYOUT mostrado en la Figura 2.26, hace referencia a un mapa completo de toda la infraestructura física del proyecto y está dividido en varias áreas (rojo), cada una de estas áreas tiene información sobre los equipos de mayor importancia que están relacionados a los registros ingresados en equipos y sistemas. La parte rodeada de azul es la que esta ampliada al momento de dar clic en el cuadrante.

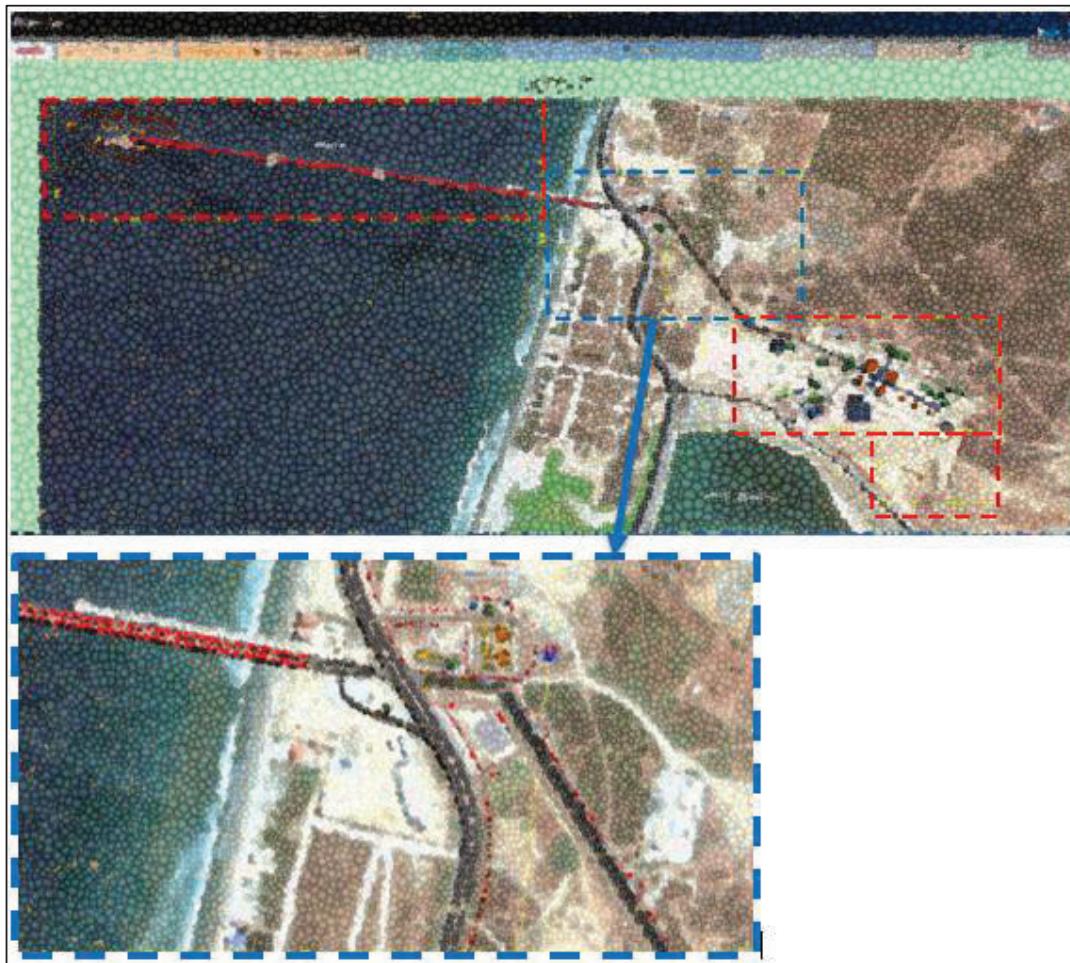


Figura 2.26: vista de consulta 3 - layout

2.4.4.4.6 Módulo de Edición

Una vez ingresado al módulo edición se muestran los siguientes vínculos de edición que se muestra en la Figura 2.27.



Figura 2.27: modulo edición

Dependiendo del tipo de información que se desee editar se seleccionará alguno de los enlaces del cuadro presentado: “Proyecto, Contratista, Disciplina, Archivo, Equipo, Sistema, Alcance o Contrato”.

Al momento de realizar la elección obtenemos el siguiente recuadro a continuación en la Figura 2.28 se coloca la identificación de cada elemento de la cabecera:



Figura 2.28: cabecera de modulo edición

Cabecera: La cabecera del menú edición cuenta con los siguientes elementos

- **Título del contenedor:** Indica el título del contenedor, mismo puede ser modificado.
- **Regresar:** Enlace para regresar al menú anterior.
- **Volver a modo consulta:** Enlace para regresar a la página principal.
- **Edición:** Enlace para ir al menú de edición de información de la base de datos.

- **Administración:** Enlace para ir al menú del ADMINISTRADOR (dicho enlace solo se muestra a usuarios que tengan los permisos respectivos).

El menú presentado “Proyectos” mostrado en la Figura 2.29 tiene cuadros de texto para llenar información como “Numeración” y “Nombre de proyecto” y también con botones de edición (inserción, listar, actualizar y eliminar) de registros.

EDICION DE REGISTROS - PROYECTOS

- Elija el proyecto - ▾ Editar

NUMERACION	<input type="text"/>
NOMBRE PROYECTO	<input type="text"/>
<input type="button" value="Nuevo"/> <input type="button" value="Listar"/>	
<input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/> <input type="button" value="Agregar"/>	

Figura 2.29: módulo de edición primario “proyectos”

El menú “Archivos” presentado en la Figura 2.30 tiene listas desplegables o combobox para elección de Proyecto, Disciplina y Alcance, también dispone cuadros de texto para llenar información como “Nombre-Especificación”, “Volumen” y “URL - Archivos” e igualmente botones de edición (inserción, listar, actualizar y eliminar) de registros.

Usuario: root

EDICION DE REGISTROS - ARCHIVOS

VOLVER A MODO CONSULTA

PROYECTO	- Elija el proyecto - ▾
DISCIPLINA	- Elija la disciplina - ▾
ALCANCE	- Elija el alcance - ▾
Nombre-Especificación	<input type="text"/>
VOLUMEN	<input type="text"/>
URL - ARCHIVOS	<input type="text"/>
<input type="button" value="Listar"/> <input type="button" value="Editar"/> <input type="button" value="Listar por Relacion"/>	
<input type="button" value="Agregar"/> <input type="button" value="Eliminar Todo"/>	

Figura 2.30: módulo de edición secundario “archivos”

2.4.4.4.7 Módulo de administrador (edición de usuarios):

El modulo administración mostrado en la Figura 2.31 solo es visible por usuarios administradores.



Figura 2.31: módulo de administrador

Como podemos divisar en la pantalla mostrada se muestra la lista actual de los usuarios creados en el sistema, se indica igualmente el tipo de usuario (verde), se tiene las opciones de editarlos o eliminarlos (rojo).

También existe la opción de crear nuevos usuarios (amarillo).

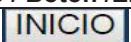
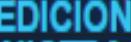
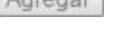
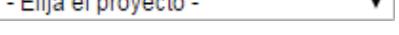
Dentro del módulo de administración mostrado en la Figura 2.31 existe un enlace denominado “LOG”, que permite al administrador observar el registro de modificaciones realizadas en el sistema por los usuarios editores y administradores, dicho sub-modulo presenta opciones de filtro: USUARIO, ACCION y RANGO-FECHA (rojo). Y de brinda opción de exportar la información obtenida en Excel.



Figura 2.32: módulo de registro de ediciones

2.4.4.5 Iconos, botones y enlaces

El sistema SGI usa los siguientes iconos y botones para el manejo, operación y administración del aplicativo Web mostrados en la Tabla 2.78, pudiendo los mismos ser usados como operadores o enlaces, a continuación se procede a detallarlos:

Descripción	Icono / Botón /Enlace
Botón de ingreso para la pantalla de autenticación: usuario/contraseña.	
Botón de ingreso al sistema.	
Enlace para Salir y cerrar la sesión actual en el sistema.	
Enlace para acceder al menú edición.	
Enlace para acceder al menú administración.	
Enlace para acceder a la ubicación de un equipo en el mapa total del proyecto.	
Enlace para regresar al modo de visualización de información y datos.	
Botón para iniciar la creación de un nuevo registro el cual se complementa con Agregar.	
Botón usado para agregar y guardar los datos del nuevo registro creado.	
Botón usado para iniciar la edición de un registro previamente seleccionado, se complementa con Actualizar y Eliminar.	
Botón para guardar la actualización del registro seleccionado.	
Botón para eliminar el registro seleccionado.	
Botón para listar todos los registros guardados.	
Combo-box usado para seleccionar un registro, se inicialmente complementa con editar y luego con actualizar o eliminar.	
Campo de texto para ingresar información	
Editar relación: Icono perteneciente al campo ACCION en el menú edición de Archivos o Equipos que permite realizar la edición de una relación creada (proyecto, alcance, disciplina).	
Botón relacionado al Icono: Editar relación, el cual permite guardar el cambio realizado en la relación.	
Guardar cambios: Icono perteneciente al menú ACCION en el menú de edición de Archivos o Equipos que permite guardar los cambios realizados a un registro. Una vez guardados los cambios el ícono cambia su color a gris indicando que ya se hicieron efectivos los cambios, y toda la fila del registro se bloquea.	 
Re-edición: Icono perteneciente al menú ACCION en el menú edición/Archivos o equipos que permite re-editar un registro que ya fue guardado desbloqueándolo y permitiendo nuevamente su edición.	

Eliminar: Icono perteneciente al menú ACCION en el menú edición de Archivos o Equipos que permite eliminar un registro. Cuando es usado automáticamente se bloquea la fila de registro indicando que el registro fue borrado. Y todo el menú ACCION se pone en un tono gris.	
Botón de avance: usado para movilizarse sobre los resultados de una consulta.	
Botón de retroceso: usado para movilizarse sobre los resultados de una consulta.	
Numeración de índice: depende de la cantidad de registros mostrados.	
Enlace de Retorno: a la página anterior	

Tabla 2.78: Descripción de iconos, botones y enlaces

2.4.4.6 Mensajes emergentes de aviso

Según las modificaciones o procedimientos que se vayan a realizar al momento de ejecutar alguna acción específica se mostrarán mensajes acordes al suceso de una acción realizada correctamente o alerta de procedimiento, a continuación se colocan los más importantes:

- La Figura 2.33 muestra el mensaje: Usted está ingresado como usuario <tipo de usuario>.

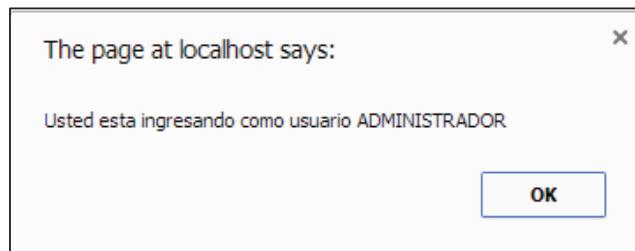


Figura 2.33: mensaje de aviso de ingreso al sistema

- La Figura 2.34 muestra el mensaje: Se insertó correctamente.

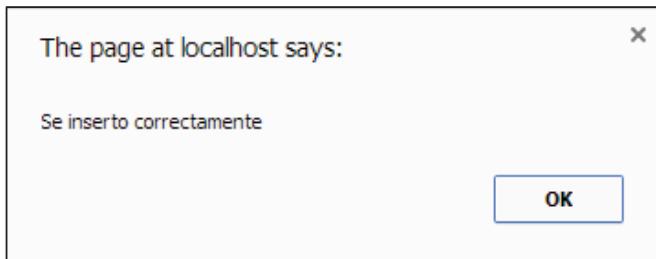


Figura 2.34: mensaje de aviso de inserción correcta

- La Figura 2.35 indica el mensaje de aviso de acción no permitida, el cual advierte al usuario sobre la imposibilidad de realización de la acción requerida, el mensaje que se muestra es el siguiente: “Hay una o más relaciones que dependen de este PROYECTO, para eliminar este proyecto debe eliminar todas las relaciones dependientes que contengan el mismo en las tablas archivos y/o equipos y/o contrato”.

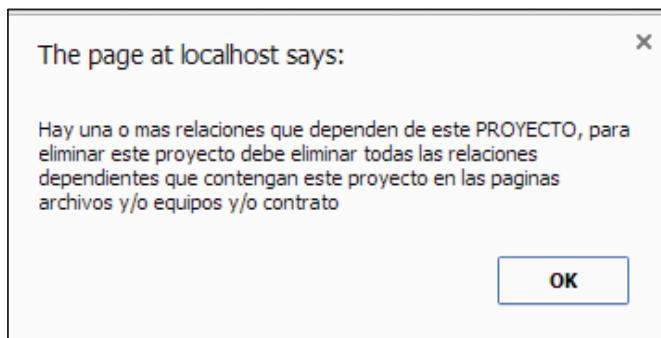


Figura 2.35: mensaje de aviso de acción no permitida.

- La Figura 2.36 muestra el mensaje de confirmación: ¿Está seguro que desea eliminar el (los) usuario (s) seleccionado (s)?.

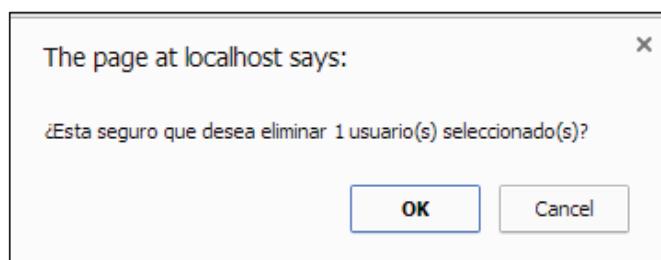


Figura 2.36: mensaje de aviso con opción a elección.

- La Figura 2.37 indica el mensaje de aviso con alerta: “El valor NOMBRE ALCANCE está vacío, no se ha guardado ningún dato”.

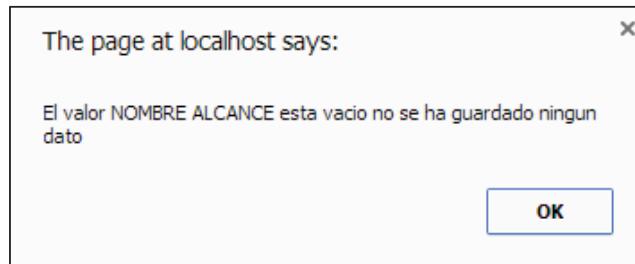


Figura 2.37: mensaje de aviso con alerta

- El mensaje de alerta de la Figura 2.38 se presenta cuando ha pasado un determinado tiempo sin ser utilizado el aplicativo Web.

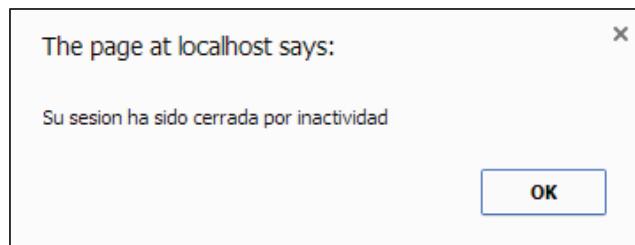


Figura 2.38: mensaje de aviso de cierre de sesión por inactividad

2.4.5 CODIFICACIÓN DE SOFTWARE

2.4.5.1 Estándares de programación

Los estándares de programación nos ayudan a generar un código fuente de mejor calidad y comprensión tanto para el desarrollador como para futuros programadores.

2.4.5.1.1 PHP

- Al escribir código php se debe usar la etiqueta <?php al iniciar se coloca el código y al finalizar se debe colocar la etiqueta ?>.

- El tipo de codificación que se debe usar al guardar el código es “UTF-8 sin BOM”, esto con el fin de evitar errores al momento de la ejecución del código.
- La concatenación e identificación de variables en php, se debe realizar usando:
 - Comillas dobles (“”): Se utiliza para que una cadena y sus respectivas variables encontradas dentro de las dobles comillas sean reconocidas.
 - Comillas simples (‘’): Se utiliza para que la cadena sea leída tal cual, sin interpretar o reconocer variables, a no ser que se agregue un concatenador de variables.
 - Puntos (..) : Es un concatenador de variables usado regularmente en cadenas de caracteres encontradas dentro de comillas simples.

2.4.5.1.2 MySQL

- Las claves primarias de la base de datos del aplicativo Web se identifican colocando el prefijo “cdg_” y el sufijo “<nombre_identificador>”.
- Los nombres identificadores de tablas dentro de la base de datos deben ir colocados con uno o dos nombres descriptivos, si son dos se colocara un guion bajo como separador entre estos.
- La clave foránea en cada tabla en donde exista se debe denominar de igual manera que la clave primaria.

2.4.5.1.3 Clases

- En determinados casos varios archivos PHP compartirán la misma clase.

- Al nombrar las clases estas deberán estar formadas por uno o varios sustantivos según sean requeridos, seguidos sin espacios y cada uno de estos deberán empezar con minúscula y solo estarán compuestos por valores del alfabeto latino.

2.4.5.1.4 Funciones y métodos:

- Las funciones y métodos solo deben contener valores del alfabeto latino.
- Los nombres identificadores de las funciones y métodos se describirán acorde a su propósito, si existe más de un sustantivo el segundo empezará sin espacios y con mayúscula.

2.4.5.2 Implementación del código de las clases del sistema

2.4.5.2.1 Clase seguridad

En la Figura 2.39 se muestra la clase seguridad la cual permite definir procedimientos de la seguridad del aplicativo Web mediante los siguientes métodos que se describen brevemente:

```

1  <?
2  class seguridad{
3      public function seguridad() {
41     public function accederModuloUsuario() {
50     public function seguridad2() {
95     public function bienvenida() {
107
108 }

```

Figura 2.39: Código de la clase seguridad

- a) **Seguridad ()**: Se encarga de validar que el usuario y la contraseña existan en la base de datos y si no existen no permite el ingreso y hace un redirecciónamiento a la página inicial. De la misma manera permite a los usuarios acceder únicamente a los módulos que tienen permiso, es decir un usuario editor no puede acceder a módulos de usuarios administrador.

- b) **accederModuloUsuario ()**: Este método o función permite al usuario según su tipo de permiso (editor o administrador) poder ingresar a los enlaces respectivos de edición y administración, a continuación se muestra el código PHP en la Figura 2.40.

```

41     public function accederModuloUsuario(){
42         if ($_SESSION["permiso_usuario"]=="1" ){
43             echo '<a href="EDICION.php" title="EDICION" style="position:absolute;
44                                         top:25px; right:10px; color: #0CF; font-size:15px;
45                                         font-family:arial;"><strong>EDICION</strong></a>';
46             echo '<a href="admin/menu_administrador.php" title="ADMINISTRACION"
47                                         style="position:absolute; top:40px; right:10px; color: #0CF;
48                                         font-size:15px; font-family:arial;
49                                         font-family:arial;"><strong>ADMINISTRACION</strong></a>';
49         }

```

Figura 2.40: Código la función AccederModuloUsuario()

Detalle:

Línea: 42 Se inicia la verificación y comparación de la variable sesión(1: Administrador, 0:Editor) del usuario actual que inició sesión, esta variable permite discriminar el tipo de usuario y respectivamente brindar acceso a los módulos correspondientes.

Línea: 43-45 Muestra el acceso a los módulos (Administración y Edición) correspondientes al administrador, los cuales son Hipervínculos. De la misma manera se establecen los parámetros visuales y de ubicación del enlace.

Línea 46: Se continúa con la verificación (ahora analiza si es un usuario Editor).

Línea 47-48: Muestra el Acceso al módulo de edición dado que la línea 46 cumpla con la condición.

- c) **Seguridad2 ()**: Método que cierra la sesión del usuario automáticamente si no hay actividad durante 5 minutos en la página. Se ejecuta mediante la ayuda de un código JavaScript.

- d) **Bienvenida ()**: Función que se encarga de mostrar el mensaje de bienvenida al iniciar la sesión en el sistema indicándole al usuario que tipo de privilegio posee.

2.4.5.2.2 Clase conexión

A continuación en la Figura 2.41 se presenta en código PHP la clase conexión que se encarga de realizar la conexión entre la base de datos MySQL y el aplicativo Web, en esta clase se definen los atributos y los constructores, estos últimos permiten inicializar los atributos y recibir parámetros. Dicha conexión es realizada mediante la función mysql_connect();

```

1 <?php
2 class conexion{
3     //SE CREAN LAS VARIABLES
4     var $servidor;
5     var $usuario;
6     var $clave;
7     var $base_datos;
8
9     public function __construct () {
10        //SE DEFINEN LAS VARIABLES
11        $this->servidor="localhost";
12        $this->usuario="root";
13        $this->clave="macros";
14        $this->base_datos="sgi";
15    }
16
17     public function conectar(){
18         global $cn;
19         //SE REALIZA LA CONEXIÓN CON EL SERVIDOR
20         $this->cn=mysql_connect($this->servidor, $this->usuario, $this->clave);
21         mysql_select_db($this->base_datos,$this->cn);
22         @mysql_query("SET NAMES 'utf8'");
23         $cn=$this->cn;
24         return($cn);
25     }
26
27 }

```

Figura 2.41: Código de la Clase conexión

Detalle:

Línea: 2 y 27 Muestran inicio y fin de la clase conexión.

Línea: 3-8 Se proceden a crear las variables para la conexión hacia la base de datos.

Línea: 9-16 Se definen los constructores de la clase, se elige a que base se conectara el usuario.

Línea: 17 Inicia la definición de la función que se usa para realizar la conexión hacia la base de datos MySQL.

Línea: 20 Se realiza la conexión hacia la base de datos y se la guarda en una variable la cual ha sido previamente referenciada.

Línea: 21 Se selecciona la base de datos a ser utilizada

Línea: 23-25 Se referencia y retorna el valor de la función para crear la conexión hacia la base de datos.

2.4.5.2.3 Clase registro

Esta clase permite editar las tablas primarias de información, tales como: alcance, proyecto, disciplina, contrato y sistemas. En la Figura 2.42 se muestra la clase con sus respectivas funciones, las cuales se describen brevemente a continuación (se colocan los más importantes):

```

1 <?
2 class registros{
3     public function __construct ( $cod="", $nom="", $sql="" ) {
4         // SE DEFINE LA VARIABLE QUE CONTIENE LA CONSULTA:MYSQL DEL REGISTRO A EDITAR
5         $this->sql;
6         // SE DEFINE LAS VARIABLES DE CONSULTA ENVIADAS DESDE OTRA PAGINA
7         $this->cod=$_POST["txt_con"];
8         $this->nom=$_POST["txt_id_con"];
9     }
10    public function valoresUnicos(){
11    public function editarRegistros($sql){
12    public function nuevoRegistro($sql){
13    public function agregarRegistros($sql){
14    public function actualizarRegistros($sql){
15    public function eliminarRegistros($sql){
16    public function listarRegistros($sql){
17    public function listarnodepRegistros($sql){
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105

```

Figura 2.42: Código de la clase registros

Detalles:

Línea: 2 y 9 Inicio y fin de la clase registros

Línea: 3-8 Se definen los constructores de la función, en donde \$sql es la variable que contiene la consulta mysql del registro que se va a editar, igualmente se definen las variables que se usarán para la consulta, mismas que son establecidas mediante el envío de un formulario de la página (anterior) de donde se realizó la petición (la respectiva llamada a la función). La variable \$cod hace referencia a la clave primaria del registro a modificar.

Línea: 10 a 97 Se definen las funciones pertenecientes a la clase registros.

a) Agregar registros primarios

- a. nuevoRegistro(): función o método mostrado en la Figura 2.43 que se encarga de asignar una numeración para el nuevo registro a ser ingresado.

```

58  public function nuevoRegistro($sql) {
59      global $cn;
60      global $var;
61      $cs=mysql_query($sql,$cn);
62      while($resul=mysql_fetch_array($cs)) {
63          $count=$resul[0];
64          $max=$resul[1];
65      }
66      if($count==0) {
67          $var="1";
68      }
69      else{
70          $var=$max+1;
71      }
72  }
73 }
```

Figura 2.43: Código de la función nuevoRegistro()

Detalle:

Línea: 58 Se define la función para crear un nuevo registro, con el argumento de entrada(el cual hace referencia a la consulta realizada a la base de datos).

Línea: 59 y 60 Se definen las variables globales que permiten que dichas variables sean usadas fuera de la clase.

Línea 61-73: Se ejecuta la consulta a la base de datos, se estable un condicional que permite que se realice la verificación de cada registro de la consulta (realizando un bucle), en donde se obtiene la cantidad de registros y el valor máximo dentro de ellos. Si el contador es igual a 0(lo que significa que no existen registros), se coloca el valor de 1(numeración), y si no se cumple esto al valor máximo obtenido se le suma el valor de 1 y así se obtiene la numeración secuencial automática.

Mediante el código mostrado en la Figura 2.44 se realiza el llamado correspondiente a la función nuevoRegistro().

```

54      //BOTON NUEVO
55      if($btn=="Nuevo"){
56          $sql="select count(cdg_proy_dis),Max(cdg_proy_dis)
57          from proyecto";
58          $clase_registros-> nuevoRegistro($sql);
59      }
60

```

Figura 2.44: Código de llamado a la función nuevoRegistro()

Detalle:

Línea: 55 Se establece un condicional el cual es ejecutado únicamente cuando se oprime un botón el cual tiene asignada un valor =“Nuevo”. Cabe recalcar que dicha valor viaja a través del método _POST.

Línea: 56 y 57 Si se cumple la condicional se ejecuta la consulta a la base de datos a la tabla proyecto en donde se obtiene el número de registros y el valor máximo de la llave primaria.

Línea: 58 Se llama a la clase registros que permite la ejecución de la función nuevoRegistro(); definida con anterioridad.

- b. El código mostrado en la Figura 2.45 permite la agregación de nuevos registros mediante el llamado a la función agregarRegistros() que completa el proceso.

```

66      //BOTON AGREGAR
67      if($btn=="Agregar"){
68          $cod=$_POST["txt_con"];
69          $nom=$_POST["txt_id_con"];
70          $sql="insert into alcance values ('$cod','$nom','')";
71          $cs1=mysql_query($test,$cn)or die(mysql_error());
72          $clase_registros-> agregarRegistros($sql);
73      }
74

```

Figura 2.45: Código de llamado a la función agregarRegistros()

Detalle:

Línea: 67 Se establece un condicional el cual es ejecutado únicamente cuando se oprime un botón el cual tiene asignada un valor =“Agregar”. Cabe recalcar que dicha valor viaja a través del método _POST.

Línea: 68 y 69 Se definen las variables que viajan por el método _POST(la clave primaria, nombre_alcance) y que serán usadas para realizar la consulta respectiva a la base de datos.

Línea: 70-74 Se realiza la inserción de los nuevos registros a la base de datos en la tabla alcance.

b) Editar registros primarios

- Este código mostrado en la Figura 2.46 permite habilitar la edición de un registro que haya sido previamente ingresado. Si no se ha seleccionado ningún registro para editar se obtendrá el mensaje: “Debe seleccionar un valor de la lista”.

```

45 //BOTON EDITAR
46 if($btn=="Editar"){
47   if($bus!='0'){
48
49     $sql="select * from alcance where nombre_alcance
      like '$bus%'";
50
51     //se envia la consulta a la clase registros
52     $clase_registros-> editarRegistros($sql);
53     while($resul=mysql_fetch_array($cs)){
54
55       $var=$resul[0];
56       $var1=$resul[1];
57
58     }else{
59       echo "<script> alert('Debe seleccionar un valor
      de la LISTA');</script>";
60     }
}

```

Figura 2.46: Código de llamado a la función editarRegistros()

c) Actualizar registros primarios

- Este código mostrado en la Figura 2.47 permite la actualización de un registro que haya sido previamente ingresado.

```

76 //BOTON ACTUALIZAR
77 if($btn=="Actualizar"){
78     $cod=$_POST["txt_con"];
79     $nom=$_POST["txt_id_con"];
80     ...
81     $sql="update alcance set nombre_alcance='$nom'
82     ',observacion_alcance='$ape1' where cdg_proy_dis='
83     $cod'";
84     $cs1=mysql_query($test,$cn)or die(mysql_error());
85     $clase_registros-> actualizarRegistros($sql);
86     ...
87 }

```

Figura 2.47: Código de llamado a la función actualizarRegistros()

Detalle:

Línea: 77 Se establece un condicional el cual es ejecutado únicamente cuando se oprime un botón el cual tiene asignada un valor =“Actualizar”. Cabe recalcar que dicha valor viaja a través del método _POST.

Línea: 78 y 79 Se definen las variables que viajan por el método _POST(la llave primaria, nombre_alcance) y que serán usadas para realizar la consulta respectiva a la base de datos.

Línea: 81-85 Se realiza la actualización de un registro existente en la base de datos en la tabla alcance (la búsqueda del registro se la realiza mediante la llave primaria). De la misma manera se llama a la clase registros la cual hará la petición de consulta de actualización mediante la función actualizarRegistros.

d) Eliminar registros primarios

- Este código mostrado en la Figura 2.48 permite eliminar los registros, los cuales inicialmente se seleccionan para luego proceder a su eliminación. Cuando una o varias relaciones dependen del registro a eliminar, el sistema no permitirá la eliminación.

```

87 //BOTON ELIMINAR
88 if($btn=="Eliminar"){
89     $cod=$_POST["txt_con"];
90     $sql="SELECT cdg_proy_dis FROM proy_disciplina
91     where cdg_proy_dis='$cod'";
92     $cs=mysql_query($sql,$cn);
93     if(mysql_num_rows($cs)==0){
94         $cs1=mysql_query($test,$cn)or die(mysql_error());
95         $sql="delete from alcance where cdg_proy_dis='
96         $cod'";
97         $clase_registros-> eliminarRegistros($sql);
98     }else{
99         echo "<script> alert('Hay una o mas relaciones que
100        dependen de este ALCANCE, para eliminar este
101        alcance debe eliminar todas las relaciones
102        dependientes que contengan este alcance en las
103        paginas archivos y/o equipos');</script>";
104     }
105 }

```

Figura 2.48: Código de llamado a la función eliminarRegistros()

Detalle:

Línea: 88 Se establece un condicional el cual es ejecutado únicamente cuando se oprime un botón el cual tiene asignada un valor =“Eliminar”. Cabe recalcar que dicha valor viaja a través del método _POST.

Línea: 89 Se definen la variable que viaja por el método _POST (llave primaria) y que serán usadas para realizar una consulta inicial antes de realizar la eliminación del registro y verificar si existen relaciones que sean dependientes del registro que se intenta eliminar.

Línea 90-100: Se realiza una consulta a la base de datos y se la almacena en la variable \$sql. Se efectúa la consulta almacenada. Se procede a verificar mediante un condicional si hay o no relaciones dependientes (0: No hay – Se procede a realizar la consulta de eliminación del registro en la base de datos, 1 o más: Se procede a mostrarle al usuario que existe 1 o más registros que dependen de la relación que se va a borrar).

e) Listar registros primarios

- A continuación en la Figura 2.49 se muestra código que realiza el listado de los registros. Inicialmente se hace la consulta en la base de datos, luego se procede a generar la tabla html con el número de columnas respectivas y se la rellena con los registros de la consulta con ayuda de un operador while.

```

156 //LISTAR
157 if($btn=="Listar"){
158     $sql="select * from alcance where nombre_alcance
159     like '%' order by cdg_proy_dis";
160     $clase_registros-> listarRegistros($sql);
161     echo "<center><table border='3'>
162         <tr>
163             <td><font
164                 color='gray'><strong>No.</strong></font></td>
165             <td><font
166                 color='gray'><strong>ALCANCE</strong></font></td>
167         </tr>";
168     while($resul=mysql_fetch_array($cs)){
169         $var=$resul[0];
170         $var1=$resul[1];
171         echo "<tr>
172             <td><font color='white'>$var</font></td>
173             <td><font color='white'>$var1</font></td>;
174     }
175     echo "</table></center>";

```

Figura 2.49: Código de llamado a la función listarRegistros()

Detalle:

Línea: 157 Se establece un condicional el cual es ejecutado únicamente cuando se oprime un botón el cual tiene asignada un valor =“Listar”. Cabe recalcar que dicha valor viaja a través del método _POST.

Línea: 158,159 Se realiza la consulta a la base de datos y se la almacena en la variable \$sql. Se llama a la clase registros para ejecutar la función listarRegistros con el argumento \$sql como parámetro de entrada.

Línea: 160-164 Se inicia la creación de la tabla donde se visualizará la consulta, la cual para este caso puntual contendrá dos campos (No. y Alcance).

Línea: 167 Se utiliza un operador condicional que imprimirá todos los resultados obtenidos (registros) de la consulta en forma ascendente, en donde \$var y \$var1 corresponden al número de campos.

Línea: 168-172 Se continúa con la creación de la tabla, aquí se imprime cada uno de los valores que generó la consulta mediante un bucle, al final se termina de establecer el formato para que se finalice la creación de la tabla.

2.4.5.2.4 Clase ingresodatos

Esta clase permite editar las tablas secundarias de información, tales como: archivos y equipos. En la siguiente Figura 2.50 se muestra la clase con sus

respectivas funciones, a continuación se describe brevemente una de las más importantes, agregarRegistros ():

```

3: class ingresodatos{
4:     public function __construct ($sql,$sql1,$sql2,$prueba="",$bus0="",$bus3="",$bus2="") {
5:         $this->sql;
6:         $this->sql1;
7:         $this->sql2;
8:         $this->prueba;
9:         $this->bus0=$_POST["txtbus0"];
10:        $this->bus3=$_POST["txtbus3"];
11:        $this->bus2=$_POST["txtbus2"];
12:    }
13:    public function paginacion($sql) {
14:    }
15:    public function agregarRegistros($prueba,$sql2) {
16:    }
17:    public function actualizarRegistros($sql) {
18:    }
19:    public function eliminarRegistros($sql) {
20:    }
21:    public function editarRegistros($sql,$sql1) {
22:    }
23:    public function listarRegistros($sql,$sql1) {
24:    }
25:    public function listarxRelacionRegistros($sql) {
26:    }
27:    public function obtenerRelacion($sql) {
28:    }
29:    public function editarRelacion($sql) {
30:    }
31:    public function ordenarRegistrosA() {
32:    }
33:    public function ordenarRegistrosE() {
34:    }
35: }

```

Figura 2.50: Código de la clase ingresodatos()

Detalle:

Línea: 3 y 146 Inicio y fin de la clase registros

Línea: 4-12 Se definen los constructores de la función, en donde los \$sql son las variables que contienen las consultas mysql del registro que se va a agregar, igualmente se definen otras variables de consulta, mismas que son establecidas mediante el envío de un formulario de la página (anterior) mediante el método _POST de donde se realizó la petición (la respectiva llamada a la función).

Línea: 13 a 130 Se definen las funciones pertenecientes a la clase registros de archivos y equipos.

Agregar registros secundarios

En la siguiente figura 2.51 se muestra el código de agregación de registros secundarios. El usuario debe elegir los datos (Proyecto, Disciplina y Alcance) del registro a ingresar, mediante el condicional if se verifica que ninguno de los datos estén vacíos, después se genera un número secuencial del id_key para cada nuevo registro.

```

55 //BOTON AGREGAR
56 if($btn=="Agregar"){
57     $nom=$_POST["txt_id_con"];
58     $url=$_POST["txt_nom"];
59     $vol=$_POST["txt_vol"];
60
61     if($bus2 == 0 || $bus3 == 0 || $bus0 == 0){
62
63     }else{
64         $prueba=mysql_query("select * from proy_disciplina where (cdg_proy_dis='
65             $bus2') and (cdg_proyecto='$bus0') and (cdg_disciplina='$bus3')");
66
67         $sql="select count(id_key),Max(id_key) from archivos";
68         $cs=mysql_query($sql,$cn);
69         $resul=mysql_fetch_array($cs);
70         $count=$resul[0];
71         $max=$resul[1];
72
73         if($count==0){
74             $var="1";
75         }else{
76             $var=$max+1;
77         }
78         //USADO PARA ACTUALIZAR EL USUARIO EN EL LOG
79         $cs3=mysql_query($test,$cn)or die(mysql_error());
80         //-----
81         $sql2="insert into archivos values ('$bus0$bus3$bus2','$nom','$vol','$url
82             ','$var')";
83         $clase_ingresodatos->agregarRegistros($prueba,$sql2);
84     }
}

```

Figura 2.51: Código de llamado a la función agregarRegistros() para la agregación de registros secundarios

Detalle:

Línea: 56 Se establece un condicional el cual es ejecutado únicamente cuando se oprime un botón el cual tiene asignada un valor =“Agregar”. Cabe recalcar que dicha valor viaja a través del método _POST.

Línea: 57-60 Se definen variables de consulta(enviadas a través del método _POST).

Línea: 61 Se aplica el condicional de verificación. Si el usuario ingreso alguno de los registros vacíos No se ejecutará la inserción. Si se da cualquier otro caso se ejecutará el código desde la Línea 64 a la 81.

Línea: 64 Se realiza la consulta y se la almacena en la variable prueba, que más adelante permitirá actualizar el registro de logs.

Línea: 66-76 Se ejecuta la consulta respectiva para generar un nuevo número de clave primaria, dicha consulta tiene la misma sintaxis y funcionamiento que la función agregarNúmero().

Línea: 78 Consulta que permite realizar la actualización y almacenamiento del cambio realizado en la tabla logs.

Línea: 80-81 Se realiza la consulta a la base de datos y se la almacena en la variable \$sql2. Se llama a la clase ingreso datos para ejecutar la función agregarRegistros() con el argumento \$sql2 como parámetro de entrada.

Una vez llamada la función agregarRegistros() que se representa en la Figura 2.52 se procede a verificar que la relación este creada y si no lo está la creará (entendiéndose por relación a la agrupación de una categoría de proyecto, disciplina y alcance en donde cada relación es única) y finalmente se genera la consulta donde se realiza la agregación del registro.

```

55 public function agregarRegistros($prueba,$sql2) {
56     global $cn;
57     $var0=$this->bus0=$_POST["txtbus0"];
58     $var3=$this->bus3=$_POST["txtbus3"];
59     $var2=$this->bus2=$_POST["txtbus2"];
60     if(mysql_num_rows($prueba)>0) {
61         $cs2=mysql_query($sql2,$cn)or die(mysql_error());
62         echo "<script> alert('Se inserto correctamente');</script>";
63     }else{
64         $sql1="insert into proy_disciplina values ('$var2','$var0','$var3','','';
65         $var0$var3$var2')";
66         $cs1=mysql_query($sql1,$cn)or die(mysql_error());
67         $cs2=mysql_query($sql2,$cn)or die(mysql_error());
68         echo "<script> alert('Se inserto correctamente');</script>";
69     }

```

Figura 2.52: Código de la función agregarRegistros()

Detalle:

Línea: 55 y 69 Inicio y fin de la función agregarRegistros(), los parámetros de entrada existentes son \$prueba y \$sql2.

Línea: 56-59 Se definen la variable de conexión y las variables de consulta (enviadas a través del método _POST).

Línea: 61-67 Se verifica si ya existen relaciones creadas que sean iguales a las que se pretende guardar actualmente, si las hay no se las crea, pero si existen se proceder a crear la relación.

2.4.5.3 CSS – Hojas de estilo

En la Figura 2.53 se describen brevemente los estilos utilizados de color, fondo, tipo de letra, tamaño de letra...etc. Que son utilizados en el aplicativo Web.

```
a { text-decoration:none }
.normal{ font-size:18px; font-family:Calibri; color:#FFF; left:15px; top: 0px; position: absolute;
}
.normal3{ font-size:17px; font-family:arial; font-style:bold; color:white; left:10px; top: 0px; position: absolute;
}
.normal1{ font-size:17px; font-family:arial; font-style:bold; color:red; right:10px; top: 10px; position: absolute;
}
.peque1{ font-size: 80px; color:#FFF; line-height: 0px; font-family:Arial;
}
.titulo { font-family:Arial; font-size: 30px; font-style: bold; color: #FFF; font-weight: normal; line-height: 0px;
}
```

Figura 2.53: Código CSS – Hojas de estilo

2.4.5.4 JavaScript - Código

En la Figura 2.54 se muestra el código JavaScript que se utiliza para dar mayor dinamismo al aplicativo Web. Se muestra un ejemplo de su utilización, la función expandida activar() se encarga de habilitar nuevamente la información de los campos de texto que se guardaron previamente, ya que al guardar estos campos se deshabilitan.

```
398 <SCRIPT language="javascript">
399 function change() {
404 function change1() {
409 function change2() {
414 function link() {
426
427 function selectChange() {
435 ...
436 function editar_user() {
451
452 function activar() {
453     document.getElementById("codigo"+arguments[0]).disabled = false;
454     document.getElementById("nombre"+arguments[0]).disabled = false;
455     document.getElementById("volumen"+arguments[0]).disabled = false;
456     document.getElementById("url"+arguments[0]).disabled = false;
457 }
458
459 function eliminar() {
488
489 function c_eliminar_todo() {
511
512 function editar_relacion(){
527 -</script>
```

Figura 2.54: Código JavaScript – Funciones

Detalle:

Línea: 398 y 527 Se define el inicio y fin del código Java Script

Línea: 399,404,409,414,427,436,459,489,512 Se definen las funciones que permitirán trabajar con la edición de registros.

Línea: 452-457 Función que permite habilitar la edición de registros (archivos o equipos) que ya han sido guardados ya que se desbloquean sus atributos de forma dinámica.

CAPÍTULO 3.

SEGURIDADES, GESTIÓN DE TRANSACCIONES Y RESPALDOS

3.1 SEGURIDADES PARA EL SISTEMA

El tema de seguridades es sumamente importante al momento de desarrollar un aplicativo Web, más aun si la información manejada o procesada dentro de este aplicativo es confidencial.

No existe una solución total para este tema pues siempre existirán vulnerabilidades en el ámbito de riesgo. En cualquier aplicativo Web implementado, el nivel de riesgo en la seguridad se incrementa conforme aumenta su interacción con el usuario y más aún si el aplicativo Web permite el ingreso o edición de datos, se tienen los siguientes casos:

- **Sistema aislado:** El caso más simple se presenta cuando el sistema únicamente está corriendo en una sola computadora y esta no está conectada a ninguna red local, en esta situación únicamente intervienen las seguridades propias del aplicativo Web.
- **Sistema compartido localmente:** Este caso se da cuando el aplicativo Web está funcionando en un equipo y el mismo está conectado a otros equipos por medio de una red local interna, en donde se puede acceder al aplicativo Web desde cada uno de ellos. Aquí se deben prever mecanismos seguridad en la transmisión de información y restricciones de acceso.
- **Sistema conectado al internet:** Finalmente se tiene el caso en el que el aplicativo se encuentra instalado dentro de una red local y adicionalmente esta está conectado al internet. En este caso como a nivel local se deben establecer mecanismos de seguridad para la protección efectiva del aplicativo Web y su información propiamente dicha.

Por lo tanto teniendo en cuenta los tres casos mencionados a continuación se procederán a indicar las medidas y acciones tomadas durante el desarrollo del aplicativo Web y los parámetros de seguridad recomendados dentro del entorno de red sobre el cual trabajará el mismo.

3.1.1 SEGURIDADES A NIVEL DEL APLICATIVO WEB

3.1.1.1 Validación de acceso a módulos y vínculos Web

El aplicativo Web desarrollado se encuentra compuesto por varias páginas de acceso o módulos, los cuales son identificados con una determinada ubicación y nombre respectivo, esto se refleja claramente en la URL, si bien es cierto al visitar un enlace específico este queda guardado en el historial de navegación, al ingresar a esa página o modulo provocando un fallo de seguridad en aplicativo, porque cualquier usuario que no pertenezca al sistema podría tener acceso a cualquier página o modulo únicamente colocando el enlace URL. Es por esto que se implementó una validación de acceso a cada página o modulo, la cual consiste en realizar una validación de datos identificando el usuario que ingreso y a su vez proporcionándole los respectivos privilegios de acceso conforme a los permisos según su perfil. Es decir que esta validación sirve tanto para el bloqueo de acceso de usuarios que quieren acceder al sistema sin iniciar sesión (al momento de colocar una URL de alguna página o modulo automáticamente es regresado a la pantalla de autenticación inicial) como para usuarios que no tienen los permisos suficientes para acceder a algún modulo o función.

3.1.1.2 Autenticación de credenciales y encriptación de contraseña

El aplicativo debe mantener la confidencialidad de las credenciales (contraseña) de acceso, las contraseñas que se guardan en la base de datos deben ser secretas (ningún usuario puede saber la o las contraseñas de uno u otro independientemente de que sea el administrador) por este motivo se ha optado por aplicar y usar un algoritmo de encriptación denominado MD5 el encripta la contraseña de modo que al momento de guardarse en la base de datos se guarda el resultado obtenido de la encriptación es decir una cadena de caracteres tipo: string de tamaño: 32 bits,

Luego al momento de la verificación de credenciales se procederá a realizar la comparación de la contraseña almacenada con la ingresada y así permitir el acceso.

3.1.1.3 Envío de variables dentro de PHP

Durante el desarrollo de la codificación del aplicativo Web se procuró hacer uso mayoritario de variables superglobales tales como `$_POST` ya que al momento de enviar información en formularios no son visualizadas por el usuario en la URL de la Web, a diferencia de las variables `$_GET`. De la misma manera se dio una mayor preferencia de uso a las variables superglobales de almacenamiento `$_SESSION` que son más seguras que las `$_COOKIES` debido a que las primeras al guardar los datos de inicio de sesión los almacenan en el servidor y no en el navegador del usuario.

3.1.1.4 Establecimiento de Timeout de sesión

Después de pasado un tiempo de inactividad en el sistema, la sesión se cerrara invalidando el identificador de sesión, con esto se logra que si algún usuario del sistema por error dejó su sesión abierta esta se cierre automáticamente después de un tiempo determinado, evitando así que quede un libre acceso a cualquier individuo ajeno por que la sesión no se cerró. Razón por la cual se procedió a implementar con la ayuda de código JavaScript, se estableció un tiempo de 5 minutos.

3.1.2 SEGURIDADES A NIVEL DE LA RED

Las seguridades a nivel de la red tienen como objetivo brindar confidencialidad, disponibilidad e integridad de la información, debido a esto se sugiere la necesidad de implementación de ciertas reglas, parámetros y configuraciones que deben cumplirse en el entorno de red donde va a funcionar el aplicativo Web. TECNIE cuenta con un servidor principal con el sistema operativo ClearOS el cual posee servicios de seguridad que deberán ser evaluados. A continuación se muestra un

esquema de la infraestructura de red de la empresa, representado en la figura mostrada a continuación:

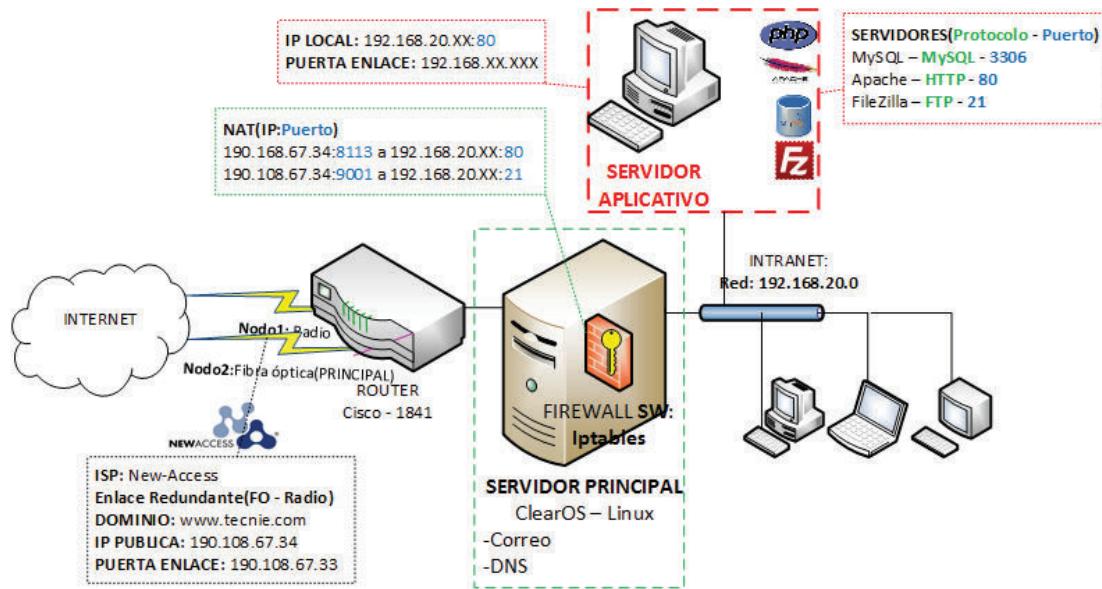


Figura 3.1: infraestructura de red de TECNIE

Como se puede observar en la Figura 3.1 un equipo de la red local es el que contiene almacenados el tanto el servidor Web Apache y el de base de datos MySQL del aplicativo Web.

3.1.2.1 Sugerencias de software de la infraestructura de red

3.1.2.1.1 Sistema operativo del servidor principal

El servidor principal del entorno de red tiene instalada una distribución Linux llamada ClearOS, este servidor cuenta adicionalmente con servicios de servidor: DHCP, Firewall, DNS y Web. También posee una interfaz de administración accesible vía Web.

Al ser este el servidor principal por el cual circulan datos internos y externos se deberán establecer y configurar las reglas respectivas para brindar seguridad al aplicativo Web, por lo que se recomienda:

- Mantener actualizado al día el repositorio de software en las últimas versiones para evitar vulnerabilidades en el sistema y reforzar su seguridad frecuentemente.
- Bloquear los puertos innecesarios que se encuentren abiertos para evitar vulnerabilidades de acceso, únicamente se necesitan tener habilitados a nivel interno los puertos 80,3306 y 21, para el servidor Web, el servidor de base de datos y el servidor FTP respectivamente.

3.1.2.1.2 Reglas para firewall software del sistema operativo

Dentro de las configuraciones del firewall de ClearOS se deben establecer las siguientes reglas:

- **WAN a LAN**, se permitirá el tráfico HTTP en el puerto 8113(se necesitará realizar port forwarding)
- **LAN**: se permitirá el tráfico HTTP en el puerto 80, se permitirá acceso al tráfico MySQL en el puerto 3306, se permitirá el tráfico FTP en el puerto 21.

El cliente desde un inicio solicita que el aplicativo y todos sus servicios se instalen en un computador asignado por ellos y ubicado en la red local interna, manifestando la idea de que desea que solo funcione a nivel local, sin embargo no descarta la idea de que también sea accesible desde el internet por lo que se necesitó establecer reglas de mapeo de puerto o Port Forwarding en el servidor para que un usuario externo mediante internet pueda acceder al aplicativo Web. Esto se resolvió mediante el uso de NAT (network address translation), que consiste en generar un vínculo/enlace entre una ip pública y un puerto externo con una ip privada y un puerto interno.

3.1.2.1.3 Reglas de port forwarding en el sistema operativo

- **WAN a LAN**: se enlazará la ip pública 190.108.XX.XX con puerto 8113 con la dirección ip privada interna 192.168.XX.XXX con puerto 80. Permitiendo

el tráfico HTTP, de la misma manera se enlazará la IP pública 190.108.XX.XX con puerto 9001 con la dirección IP privada interna 192.168.XX.XXX con puerto 21. Permitiendo el tráfico FTP.

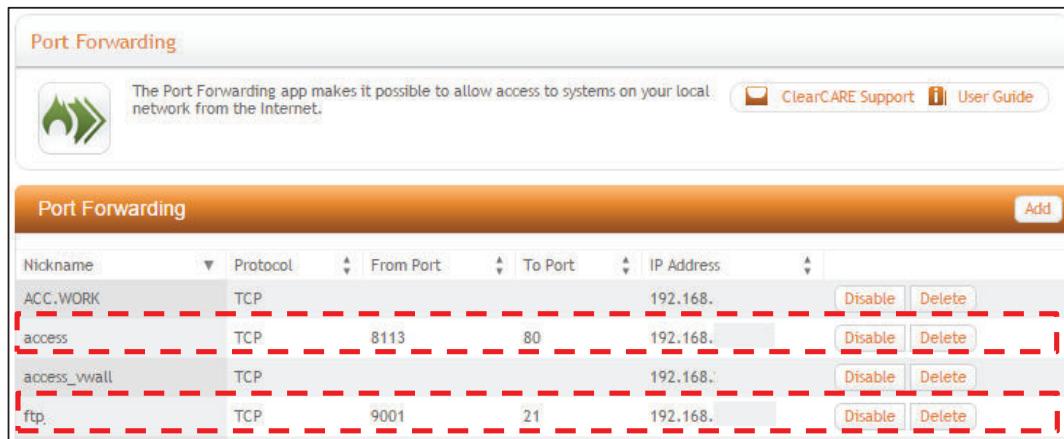


Figura 3.2: creación de reglas de port forwarding

3.1.2.1.4 Dispositivos de conexión inalámbrica interna

Actualmente en la red interna existen dos Access Point, se recomienda que ambos equipos cuenten con un tipo de seguridad WPA2, esto en función de evitar que cualquier usuario ajeno ingrese a la red local interna.

3.1.3 RECOMENDACIONES PARA SOFTWARE DEL APLICATIVO

3.1.3.1 Apache Server y PHP

Se indican las siguientes recomendaciones.

3.1.3.1.1 Ocultamiento de información del software del aplicativo

Ocultar versión de apache, php y mysql y demás servicios que intervienen en el funcionamiento del aplicativo debido a que usuarios ajenos podrían usar esta información en su beneficio, buscar vulnerabilidades y realizar algún ataque a estos servidores o servicios.

3.1.3.1.2 Bloquear el acceso directo a archivos mediante URL

De la misma manera tal como se mencionó en los enlaces URL de los módulos o páginas del aplicativo Web, cualquier usuario ajeno o externo que conozca la dirección URL específica de un archivo almacenado en el servidor Web (logs – historial de un navegador), tan solo con colocar la dirección URL exacta del archivo bastaría para descargar y obtener el archivo.

Para evitar esto se debe realizar una modificación en el archivo de configuración del servidor Web Apache denominado .htaccess usado para especificar restricciones de seguridad. Por lo que se agregan las siguientes líneas al archivo de configuración mencionado:

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^http://localhost.*$ [NC]
RewriteCond %{HTTP_REFERER} !^http://localhost*$ [NC]
RewriteRule ^\.(jpg|jpeg|gif|png|bmp|zip|pdf)$ - [F,NC]
```

Cada vez que el cliente hace una solicitud en el servidor Web, en esta solicitud viajan los datos de los encabezados o HTTP headers que contiene información del aplicativo que está haciendo la petición, por lo que si la petición es de un lugar externo al sitio del servidor se denegara el acceso. El usuario obtendrá un mensaje de acceso denegado “403 Forbidden”.

3.1.3.2 MySQL Server y PHP

3.1.3.2.1 Cambio de contraseña de del usuario root en MySQL(durante la instalación)

Establecimiento/cambio inmediato de la contraseña del usuario root luego de su instalación (cuando sea llevada a cabo).

3.1.3.2.2 Creación de un usuario en MySQL con permisos restringidos

Creación de un usuario con permisos restringidos en archivo “conexión.php” para la conexión con la base de datos para no usar el usuario root.

3.1.3.2.3 Restricción de acceso por IP

Restricción de acceso solo a conexiones autorizadas de direcciones de red específicas configuradas en el servidor MySQL, esto se logra colocando la dirección 127.0.0.1 en el host., se hace uso la utilidad PHPMyAdmin, se muestra la Figura 3.4 a continuación.

User	Host	Password	Global privileges	Grant	
<input type="checkbox"/> Any	%	—	USAGE	No	
<input type="checkbox"/> Any	localhost	No	USAGE	No	
<input type="checkbox"/> Any	production.mysql.com	No	USAGE	No	
<input type="checkbox"/> root	127.0.0.1	No	ALL PRIVILEGES	Yes	
<input type="checkbox"/> root	localhost	Yes	ALL PRIVILEGES	Yes	
<input type="checkbox"/> root	production.mysql.com	No	ALL PRIVILEGES	Yes	

Figura 3.3: configuración para restringir acceso a la base de datos desde PHPMyAdmin

3.1.4 RECOMENDACIONES PARA ASEGURAMIENTO DE RED FÍSICA

Se sugiere y recomienda lo siguiente:

3.1.4.1 Delegación y asignación de Recurso humano apto

- La persona que se elija como administrador de red deberá poseer los suficientes conocimientos para implementación, mantenimiento y resolución de problemas presentados en el aplicativo Web y en la red.
- La instalación de software del aplicativo Web debe ser realizada solo por personal designado.

3.1.4.2 Administrativo: Definición de procedimientos y Contraseñas robustas

- Es necesaria la definición de procedimientos de backup y recuperación ante fallos de los servidores del aplicativo Web, ya sea de tipo software o hardware.

- Las contraseñas manejadas en aplicativos y servicios de alta importancia deben ser bastante robustas, especialmente si estos aplicativos son usados por el administrador.
- Es necesaria la definición de políticas de revisión de bitácoras y registro de eventos en los servidores para solventar problemas de fallos o errores presentados.

3.1.4.3 Seguridad física

Debe existir seguridad física para el servidor principal de la empresa, solo personas autorizadas pueden obtener acceso al mismo, en lo posible se debe usar cerradura.

3.1.4.4 Análisis de vulnerabilidades y plan de contingencia ante fallos

La infraestructura red donde se probará el aplicativo Web generado deberá ser evaluada en función de verificar su correcto desempeño y seguridad.

Se verificarán vulnerabilidades, brechas de seguridad y parámetros de configuración para la corrección de las mismas estableciendo procedimientos y estrategias de aseguramiento, por lo que se realizará la preparación de un plan de contingencia.

3.1.4.5 Preparación de un plan de contingencia

Independientemente de que existan vulnerabilidades (externas o internas) o que el sistema falle se debe tener un plan de contingencia ante cualquier eventualidad. De modo que se puede dar una rápida respuesta a estos eventos no previstos.

3.2 GESTIÓN Y MONITOREO DE TRANSACCIONES

Al ser este un aplicativo Web que funciona directamente con una base de datos se hace necesario el monitoreo de las transacciones de información realizadas por usuarios, dicha gestión de transacciones es necesaria para:

- Verificar y detectar posibles errores o fallas en el sistema que puedan suscitarse. El tener conocimiento del último registro editado antes de un fallo permitirá a hacer un análisis del error y buscar una solución más rápida.
 - Funcionar como un mecanismo de auditoría de edición de registros de la base de datos, en donde se almacenará el tipo de cambio realizado, el usuario que lo implementó y la fecha del suceso.

Debido a los puntos mencionados el aplicativo cuenta con un sistema interno de monitoreo de transacciones al que tienen acceso únicamente los usuarios administradores. Esto se lo realizó con ayuda de triggers configurados en MySQL, los triggers o disparadores son procedimientos que se auto ejecutan cuando se cumple una condición que se estableció previamente al generar un operación (añadir, actualizar y eliminar). A continuación en la Figura 3.5 se coloca un ejemplo de la creación de triggers realizados para la tabla “Proyecto” que permite guardar el registro de los cambios que el usuario ha realizado en la base de datos.

Figura 3.4: ejemplo de código de implementación de trigger.

Cada vez que se realiza alguna edición, los triggers se ejecutan automáticamente lo que hace que dichos registros de transacción se almacenen en una tabla llamada “logs” de la base de datos “sgl_log”, que servirá como una bitácora de edición de

datos y servirá para buscar posibles fallas y errores que se den en el aplicativo Web (base de datos). Se coloca a continuación un gráfico del módulo logs, representado en la Figura 3.5.



Figura 3.5: módulo de logs

3.3 RESPALDOS DE BASES DE DATOS

El respaldo de las bases de datos del sistema es crucial teniendo en cuenta la posibilidad de que exista un evento de fallo tanto de hardware como software, por lo que debe existir un mecanismo que asegure la disponibilidad y rápida recuperación de datos. Razón por la cual se ha implementado en el equipo donde se instaló el aplicativo Web y su base de datos un mecanismo de respaldo automático de los archivos de las bases de datos mediante scripts batch.

3.3.1 CREACIÓN DE UN ARCHIVO BATCH

Un archivo batch es un archivo de texto plano que contiene órdenes de ejecución, y posee una extensión .bat, por lo que se ha creado uno para que se realice el respaldo automático de todas las bases de datos del aplicativo Web. En la Figura 3.6 se muestra el archivo .bat implementado.

```
set FECHA=%date%
set FECHA=%FECHA:/=%
set FECHA=%FECHA: =%
set FECHA=%FECHA::=%
set FECHA=%FECHA:,=%
set FILE=C:\\Backups\\Backup_%FECHA%.sql
C:/AppServ/MySQL/bin/mysqldump.exe -hlocalhost -u . -p . -r %FILE% sgi
```

Figura 3.6: código de archivo batch

Una vez realizado este archivo se procede crear la regla de ejecución:

- **Para Windows XP:** Ir a panel de control ->Tareas Programadas.
- **Para Windows 7,8 o superior:** Ir a Herramientas del sistema ->Programador de tareas.

Se crea una nueva tarea y se establece una periodicidad de ejecución, finalmente se le asigna la ejecución del archivo .bat.

3.3.2 PROGRAMACIÓN DE BACKUP DE BASES DE DATOS MANUAL

También es necesario realizar un backup manual de la base de datos, puede ser semanal o mensual dependiendo de la necesidad, para esto se utiliza la herramienta phpMyAdmin, mediante la cual fácilmente se puede hacer el backup de todas las bases de datos.

Pasos a seguir para realizar el Backup manual:

- Ingresar a la interfaz Web de phpMyAdmin elegir la base objetivo a respaldar, buscar la opción exportar, seleccionar el tipo de formato en el que queremos el respaldo (.sql), elegimos las opciones según nuestra conveniencia, damos clic en Continuar, como se muestra en la Figura 3.7.

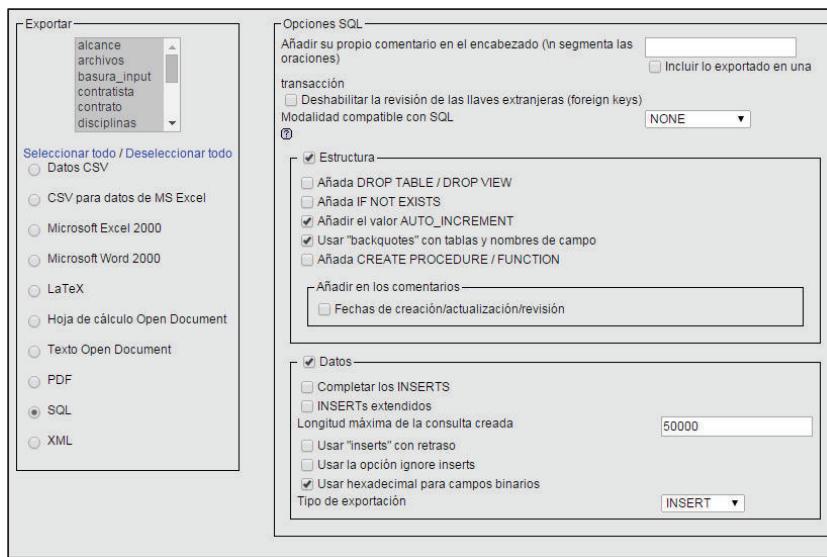


Figura 3.7: exportar código MySQL desde PHPMyAdmin

- Inmediatamente nos aparece un cuadro de texto mostrado en la Figura 3.8 de la base de datos.
- Procedemos a copiar todo el contenido encontrado en el cuadro de texto de MySQL en un bloc de notas y lo denominamos con el nombre y la fecha de respaldo y lo guardamos con extensión .sql.

```

Servidor: localhost ▶ Base de datos: sgi
-- phpMyAdmin SQL Dump
-- version 2.10.3
-- http://www.phpmyadmin.net
--
-- Servidor: localhost
-- Tiempo de generación: 14-01-2015 a las 22:24:20
-- Versión del servidor: 5.0.51
-- Versión de PHP: 5.2.6

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

-- 
-- Base de datos: `sgi`
--

-- 
-- Estructura de tabla para la tabla `alcance`
--


CREATE TABLE `alcance` (

```

Figura 3.8: archivo de texto plano, código sql

De esta manera se tendrá un repositorio de archivos de backup de la base de datos del sistema y cada vez que se necesite hacer alguna **migración o restauración** de la base de datos, simplemente se elegirá el archivo con el código SQL guardado a la fecha DD-MM-AAA y se procederá a cargar la nueva información de la base de datos.

Pasos a seguir para restaurar el backup:

Se ingresa al Shell de la base datos de MySQL, damos clic en botón inicio, luego clic en botón ejecutar, se ingresa el texto “cmd”, se coloca mysql y se oprime Enter. Dentro del Shell de mandatos se crea la base de datos “sgi” con el comando:

- **create database sgi;**

Luego de esto se copia todo el texto guardado en uno de los archivos de backup realizados y se lo pega en el Shell de MySQL, se oprime la tecla Enter. Y así se recuperaran todos los registros ingresados a la fecha del backup.

CAPÍTULO 4.

PRUEBAS, ANÁLISIS DE RESULTADOS Y COSTOS

4.1 PRUEBAS DEL SOFTWARE

Según la metodología escogida se deben realizar dos tipos de pruebas: unitarias (ya fueron realizadas durante el proceso de codificación) y de aceptación (realizadas por el cliente). Esto con el fin de evaluar si se cumplieron los objetivos y requerimientos planteados en la etapa inicial del proyecto.

Equipo de desarrollo y pruebas unitarias del aplicativo Web: Se instalaron los servidores y herramientas necesarias para implementar el sistema.

Características del equipo:

- Tipo: PC – Monitor 17"
- Procesador: Pentium 4 CPU 1.5Ghz
- Memoria Ram: 2GB
- Sistema Operativo: Windows XP SP3
- Tipo de sistema: 32 bits
- Disco duro: 250GB

Equipo de presentación definitiva del aplicativo Web: Se instalaran los servidores respectivos para la ejecución del aplicativo, en el cual se harán las pruebas de aceptación con el usuario que al final será el equipo donde se entregará formalmente el aplicativo.

Características:

- Tipo: AIO (All-in-One) – Touch Screen 21"
- Procesador: Intel Core(TM) i3-3220 CPU 3.30Ghz
- Memoria Ram: 4GB
- Sistema Operativo: Windows 8

- Tipo de sistema: 64 bits
- Disco duro: 1000GB

4.1.1 PRUEBAS DE ACEPTACIÓN

Estas pruebas tienen como objetivo verificar si las historias de usuario realizadas durante la etapa de codificación se cumplieron a cabalidad, así como determinar si el sistema desarrollado es aceptado por los usuarios o clientes, dando un punto de vista y criterio sobre la funcionalidad y rendimiento.

Las pruebas son preparadas por el desarrollador y ejecutadas por el usuario o cliente final.

Las pruebas de aceptación serán realizadas en el equipo de cómputo que se indicó al inicio de este capítulo.

Las pruebas fueron realizadas en los siguientes ambientes:

- Equipo de cómputo.
- Red local (cableada).
- Internet.

Se detallan las características de la plantilla que fue usada para realizar las pruebas:

- **Código:** Es la notación que diferencia y numera a las pruebas de aceptación.
- **Historia de usuario:** Es el código de la historia de usuario a evaluar.
- **Tarea de usuario:** Es el código de la tarea de usuario a evaluar.
- **Nombre:** Es el nombre de la historia de usuario a evaluar.
- **Descripción:** Es el nombre de la tarea de usuario a evaluarse.
- **Condiciones previas de ejecución:** Son las pre-condiciones que deben existir antes de que se ejecute la prueba o evaluación.

- **Pasos (Ejecución):** Son los pasos a seguir para ejecutar la prueba de aceptación.
- **Resultado esperado:** Es el resultado obtenido de la prueba de aceptación.
- **Evaluación:** Es el nivel de satisfacción del usuario que realizó la prueba.

A continuación se presentan las pruebas de aceptación realizadas:

Código:	PA1
Historia de usuario:	HU1
Tarea de usuario:	T1
Nombre:	Acceso al sistema
Descripción:	Acceder al sistema por medio login
Condiciones(de ejecución previas:	<ul style="list-style-type: none"> • El usuario debe estar creado en el sistema. • La contraseña y el usuario deben ser correctos.
Pasos (Ejecución):	<ul style="list-style-type: none"> • El usuario se conecta al sistema con un navegador Web. • Dar clic en inicio. • Ingresar nombre de usuario y la contraseña otorgada el administrador. • Dar clic en iniciar sesión.
Resultado esperado:	<ul style="list-style-type: none"> • Se logró ingresar con éxito al menú principal. • Se le presentaron las opciones y menús correspondientes al tipo de usuario. • Se probó ingresando datos falsos: usuario y contraseña, no fue posible el acceso.
Evaluación:	El cliente calificó la prueba como satisfactoria y recomendó: Indicarle al usuario que ingresa a la base que tipo de usuario es por medio un mensaje.

Tabla 4.1: Prueba de aceptación 1

Código:	PA2
Historia de usuario:	HU2
Tarea de usuario:	T1
Nombre:	Gestión de contenedor: proyecto
Descripción:	Crear nuevo registro
Condiciones(de ejecución previas:	<ul style="list-style-type: none"> • El usuario debe existir en el sistema. • El usuario debe tener permisos de editor o administrador. • El usuario debe haber iniciado sesión en el sistema.
Pasos (Ejecución):	<ul style="list-style-type: none"> • El usuario ingresa al sistema, coloca sus datos: usuario y contraseña e inicia sesión. • Dar clic en el enlace EDICION ubicado en la esquina superior derecha. • Dar clic en el enlace PROYECTO ubicado en el cuadro de Ingreso de información. • Dar clic en el botón Nuevo. • Colocar en el campo de texto “NOMBRE DEL PROYECTO” el valor correspondiente. • Dar clic en el botón Agregar.
Resultado esperado:	<ul style="list-style-type: none"> • Se logró ingresar con éxito al menú principal y después al menú edición.

	<ul style="list-style-type: none"> • Al momento de dar clic en el botón Nuevo automáticamente apareció un número en el campo de texto NUMERACION. • Al momento de dar clic en el botón Agregar para guardar el registro se obtuvo un mensaje “El registro se insertó correctamente”.
Evaluación:	El cliente califico la prueba como satisfactoria y no brindo comentarios al respecto.

Tabla 4.2: Prueba de aceptación 2

4.1.2 EVALUACIÓN DE RESULTADOS

4.1.2.1 Velocidad de transacciones

Las pruebas de aceptación fueron realizadas en tres casos distintos (local, red interna e internet), la principal diferencia del procesamiento en los distintos ambientes radicó en la velocidad de respuesta de las transacciones: menos de 1 segundo en el equipo de cómputo, 2 segundos en la red local (cableada) y de 3-4 segundos por internet. Se logró detectar un problema de lentitud al momento de querer acceder, visualizar o descargar archivos PDF debido a que se manejan tamaños entre los 75MB y 100MB en todos los proyectos, por lo que se recomendó al cliente utilice alguna herramienta de división de archivos para disminuir el tamaño antes de subirlos al servidor ftp.

4.1.2.2 Concurrencia: Acceso al sistema e ingreso de información

El sistema puede tener como máximo la cantidad de 10 usuarios concurrentes (esta configuración se implementó en el archivo de configuración de MySQL - my.ini), sin embargo se realizó la prueba para 5 usuarios simultáneos sin complicaciones en donde se procesó la edición de información.

Con estos aspectos y algunas sugerencias sobre modificaciones pequeñas pedidas del cliente se dieron como concluidas las pruebas de aceptación de manera positiva.

4.2 ANÁLISIS DE RESULTADOS Y COSTOS DE LA APLICACIÓN

4.2.1 RESULTADOS DE LA APLICACIÓN

En función de obtener una perspectiva global del aplicativo Web desarrollado e implementado, se ha procedido a realizar un cuestionario dirigido a los usuarios de la empresa TECNIE que han usado el aplicativo Web, el mismo que se lo ha realizado de la manera más simple en función de que sea fácil de analizar. Se ha optado por usar el cuestionario SUS (System Usability Scale), indicado a continuación.

4.2.1.1 SUS (System Usability Scale)^[46]

Es una herramienta válida para la medición y evaluación de usabilidad en software desarrollado en este caso el aplicativo Web, este cuestionario fue creado en el año de 1986 por Jonh Brooke, mismo que puede ser muy fácil de usar e interpretar.

4.2.1.1.1 Forma de cálculo

A continuación se indica la forma de cálculo y evaluación del cuestionario S.U.S.:

Existen 10 preguntas, cada pregunta tiene un valor de 1 a 5 según le otorgue el usuario. La obtención del SUS en el caso de las preguntas 1, 3, 5, 7 y 9 se la hace restando el valor otorgado por el usuario menos 1. La obtención de SUS en caso de las preguntas 2, 4, 6, 8 y 10 se la hace restando 5 menos el valor de la pregunta otorgado por el usuario. El resultado se multiplica por 2.5 para obtener el resultado final.

A continuación se presentan las preguntas de la plantilla del cuestionario S.U.S.

Preguntas:

1. Me gusta usar frecuentemente este software?
2. Encuentro el software innecesariamente complejo?
3. Pienso que es un software fácil de usar?

4. Pienso que necesito soporte de algún técnico para manejar el software?
5. Encontré varias funciones en el sistema que fueron muy bien integradas?
6. Pienso que hay muchas inconsistencias en el software?
7. Imagino que la mayoría de gente aprenderá a utilizar este sistema rápidamente?
8. Me pareció un sistema muy complicado de usar?
9. Me siento muy seguro al usar el sistema?
10. Necesito aprender bastantes cosas antes manejar el software?

Preg. No.	Usuario 1	Cálculo	Usuario 2	Cálculo	Usuario 3	Cálculo	Usuario 4	Cálculo
1	3	2	3	2	2	3	3	2
2	0	5	1	4	2	3	2	3
3	4	3	4	3	3	2	4	3
2	1	4	2	3	3	2	3	2
5	3	2	4	3	2	1	3	2
6	1	4	1	4	1	4	0	5
7	4	3	5	4	5	4	5	4
8	1	4	1	4	2	3	3	2
9	4	3	4	3	4	3	3	2
10	2	3	0	5	3	2	1	4
SUS		33		35		27		29
% satisfacción		82,5		87,5		67,5		72,5

Tabla 4.41: Matriz de evaluación de software -SUS (System Usability Scale)

De las cuatro personas encuestadas y sus respectivos resultados promedio representados en la Figura 4.41 el resultado promedio de satisfacción y aprobación global es de **77,5%**, lo cual indica que la usabilidad del aplicativo Web está dentro de un rango aceptable.

4.2.2 COSTOS DE LA APLICACIÓN

Para la valoración del costo total del software se han tomado los siguientes costos directos:

4.2.2.1 Costos de hardware

Se refiere al equipo computacional necesitado para desarrollar el aplicativo y para la presentación del mismo, en este caso ambos equipos ya pertenecían con antelación a la empresa, por lo que no fue necesario hacer una inversión. Costo \$0.00.

4.2.2.2 Costos de Software para desarrollo

Hace referencia al trabajo, creación de código y programación con herramientas de software licenciado. No existe porque todo el software usado es libre Costo \$0.00.

4.2.2.3 Costos de licencia para desarrollo

Implica si el trabajo se realizó sobre una tecnología con licencia. No existe, debido a que no se necesitó ninguna licencia. Costo \$0.00.

4.2.2.4 Costos de desarrollo

Es el tiempo invertido por el desarrollador para implementar el software y sus respectivas pruebas, se lo valora mediante hora de desarrollo. El valor establecido es de \$6.00 la hora.

4.2.2.5 Costos de aprendizaje

Es el tiempo invertido por el desarrollador en la investigación del conocimiento necesario para implementar el software, incluyendo todo el análisis respectivo y la captación de datos. El valor establecido es de \$7.00 la hora.

4.2.2.6 Cálculo de costos

Para cálculo de costos del aplicativo Web mostrados en la Tabla 4.42 se multiplican los costos directos por la cantidad efectiva de horas empleadas. Las horas efectivas fueron tomadas del cuadro de los tiempos establecidos para el desarrollo de las historias de usuario que suman un total 531 horas efectivas.

El costo total llega aproximadamente a ser \$2900.00, lo que claramente representa un ahorro a la empresa debido a que solo son gastos netamente de desarrollo y no de compra o alquiler de software o tecnología.

Costos Directos	Valor(\$)/Hora	Valor(\$)	Cantidad Efectiva(horas)	Total(\$)
Hardware		\$0.00		\$0.00
Licencia de desarrollo		\$0.00		\$0.00
Software de desarrollo		\$0.00		\$0.00
Aprendizaje e investigación	\$6.00		200	\$1200.00
Desarrollo	\$5.00		331	\$1655.00
TOTAL			531	\$2855.00

Tabla 4.42: tabla de costos

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

El nivel de aceptación por parte del cliente luego de usar el aplicativo Web fue del 77.5%, analizando estos resultados se puede concluir que las expectativas implementación planteadas se cumplieron en gran mayoría, sin embargo siempre pueden existir mejoras en la funcionalidad, y la mejor forma de aplicarlas es en el sitio con el cliente que día a día hace uso del aplicativo por lo que es ahí cuando salen los nuevos requerimientos y la necesidad de mejoras, con esto se puede llegar a una satisfacción completa del cliente.

Conforme se fue desarrollando la codificación del aplicativo Web acorde a la metodología XP y las historias de usuario propiamente dichas se fueron presentando nuevos requerimientos por parte del cliente, lo que claramente muestra que es necesaria la realización de un constante trabajo de retroalimentación directo y repetitivo con el mismo para la optimización y compleción del aplicativo.

Con este aplicativo ahora es posible acceder y trabajar remotamente desde cualquier lugar siempre y cuando exista una conexión a internet, en las ocasiones que sea necesario y autorizado por la empresa.

La falta de experiencia en el desarrollo de este tipo de aplicaciones Web usando y siguiendo metodologías de desarrollo hizo que el proyecto demore más de lo planificado por que existieron complicaciones de varios tipos, sin embargo es innegable el hecho de que seguir una metodología para el desarrollo de software ayuda a conseguir un mejor y más robusto producto.

La universalidad del uso del aplicativo mediante vía Web es bastante ventajosa, la tendencia del cloud computing hace que las empresas prefieran el desarrollo de software tipo Web debido a su facilidad de acceso, pensando en esto se creó el

aplicativo SGI en un entorno Web puesto que no necesita instalarse ningún software o complemento para funcione, únicamente es necesario un navegador Web (Google Chrome) y el explorador de Windows para su ejecución y utilización.

El desarrollo e implementación del sistema fue realizado con herramientas de software libre y con equipo de computacional existente pertenecientes al cliente, lo que proporciona un ahorro en costos ya que únicamente se generaron costos propiamente de investigación y desarrollo.

La implementación de la gestión de transacciones de la base de datos del aplicativo Web llamada “LOGS” es una herramienta que tiene dos usos de peso:

1. Detecta errores o fallos suscitados en el aplicativo, debido a que se mantiene un registro de edición con fecha de los procedimientos realizados.
2. Realiza auditorías de la edición de información en el aplicativo que son hechas por usuarios en la base de datos.

La capacitación e inducción sobre el uso y funcionamiento del sistema para los usuarios editores y administradores es totalmente necesaria ya que si bien es cierto que se establecieron reglas, mensajes de aviso y alertas, siempre puede existir algún nuevo problema ocasionado por mal manejo por parte de los usuarios que provoque errores o fallos en el aplicativo Web, por lo que es necesario tener una capacitación completa del funcionamiento del sistema.

La existencia de una amplia cantidad información sobre programación en PHP, MySQL y javaScript ayudó bastante en el momento de la codificación o programación, puesto que se presentaron algunas dudas y situaciones complicadas.

La técnica de spikes o bosquejos utilizada en el desarrollo de la interfaz gráfica fue de gran éxito al momento de desarrollo de las plantillas mediante hojas de estilo CSS ya que el cliente las propuso a su criterio y este quedo satisfecho porque al final el resultado fue bastante similar al solicitado.

El desarrollo e implementación del aplicativo Web representa para la empresa una herramienta sólida tanto de control, ordenamiento, almacenamiento y consulta de archivos, lo que permite tener mejor control de calidad en la documentación recabada de un cierto proyecto que pertenece al dossier de calidad/construcción/ingeniería.

El uso del aplicativo Web permite agilitar la preparación y elaboración del producto final (creación de la biblioteca de consulta) siempre y cuando se tenga clara la estructura del contenido y se empiecen a receptar los archivos técnicos digitales finales de la compleción del proyecto que irán vinculados al sistema. Como resultado de esto se obtiene una mayor eficiencia y profesionalismo en la entrega del producto final al cliente con el que trabaje TECNIE.

Según la metodología XP las historias de usuario dentro del desarrollo e implementación del aplicativo Web tienen que ser lo más simples posibles, por lo que se debe procurar buscar varias maneras de cumplir lo solicitado mediante la codificación del código y si se encuentran problemas o demasiada complejidad en su desarrollo se deben buscar nuevos caminos o alternativas para así evitar perder el tiempo de desarrollo.

La facilidad de visualización para un usuario lector es tal que cualquier tipo de usuario con conocimientos básicos de internet y de cualquier profesión entendería y aprendería fácilmente su funcionamiento.

5.2 RECOMENDACIONES

Se recomienda realizar proyectos de este tipo al menos entre dos personas de modo que exista un apoyo mutuo especialmente en la parte de codificación que es la más complicada, ya que siempre es bueno tener más de un criterio al momento de resolver temas como este.

Se recomienda dictar como catedra optativa la materia denominada ingeniería de software, la cual servirá de gran ayuda a futuros desarrolladores de proyectos similares al presentado en esta tesis, porque cuando se tienen bases sólidas previas sobre la materia investigada el proyecto a implementarse se desarrolla más rápido y mejor, y por ende esto ayuda a que la efectividad y eficiencia en implementación y desarrollo de este tipo de proyectos sea más alta.

Se recomienda al usuario con permisos de edición que vaya a utilizar este aplicativo Web lea su respectivo manual y en lo posible esté presente en una clase tutorial introductoria dictada por desarrollador o administrador del sistema. Ya que se necesitan manejar ciertos criterios y conocimientos para el correcto uso de la aplicación.

Se recomienda seguir en lo posible las sugerencias indicadas en el capítulo 3 de este proyecto sobre las seguridades que debería tener el entorno de red donde funcione el aplicativo, esto con el fin de aumentar la confiabilidad y disponibilidad del sistema.

Se recomienda al administrador del aplicativo que se mantenga al día en los conocimientos sobre gestión y mejoras de seguridad tanto de la infraestructura de red como del aplicativo de modo que a futuro pueda actualizar dichas seguridades acorde a los avances de las tecnologías y así evitar cualquier percance que se pueda presentar.

Se recomienda tomar muy en cuenta la definición y realización de procedimientos de rápida recuperación contra fallos del sistema, mismas que deben ser documentadas y estar listas a la mano para la solución de fallos leves o graves presentados, y así no perder la continuidad de trabajo.

En la etapa de pre-desarrollo se recomienda enseñar breves conceptos de inducción a los usuarios del aplicativo Web sobre la tecnología a utilizarse y las bondades de la misma, de modo que exista un mejor nivel de interactividad sobre

los requerimientos del cliente objetivo y así pueda tener más claro lo que se va a desarrollar.

Se recomienda evitar colocar información o campos de texto innecesarios en la interfaz gráfica en la que trabaja el usuario ya que solo hacen que el usuario se confunda.

REFERENCIAS BIBLIOGRÁFICAS

TEXTOS

- [1] CÁCERES Paloma, Procesos ágiles para el desarrollo web. Madrid: Grupo Kybele, 2001.
- [2] GARZAS, J. (2013). Metodologías Ágiles. Obtenido de <http://www.javiergarzas.com/metodologias-agiles>
- [14] COBO Angel, PHP y MySql.tecnologías para el desarrollo de aplicaciones web. Editorial Diaz de Santos. Madrid. 2005,pp 3.
- [15] COBO Angel, PHP y MySql.tecnologías para el desarrollo de aplicaciones web. Editorial Diaz de Santos. Madrid. 2005,pp 4,5.
- [16] COBO Angel, PHP y MySql.tecnologías para el desarrollo de aplicaciones web. Editorial Diaz de Santos. Madrid. 2005,pp 6.
- [17] MORA S. L., Programación en internet:Cientes Web, 2011.
- [18] AMAYA J. A., Sistemas de Información Gerencial, Universidad Santo Tomas.
- [19] COBO Angel, PHP y MySql.tecnologías para el desarrollo de aplicaciones web. Editorial Diaz de Santos. Madrid. 2005.
- [21] COBO Angel, PHP y MySql.tecnologías para el desarrollo de aplicaciones web. Editorial Diaz de Santos. Madrid. 2005, pp 12.
- [22] COBO Angel, PHP y MySql.tecnologías para el desarrollo de aplicaciones web. Editorial Diaz de Santos. Madrid. 2005, pp 13.
- [23] COBO Angel, PHP y MySql.tecnologías para el desarrollo de aplicaciones web. Editorial Diaz de Santos. Madrid. 2005, pp 14.
- [25] COBO Angel, PHP y MySql.tecnologías para el desarrollo de aplicaciones web. Editorial Diaz de Santos. Madrid. 2005, pp 16.
- [26] COBO Angel, PHP y MySql.tecnologías para el desarrollo de aplicaciones web. Editorial Diaz de Santos. Madrid. 2005, pp 17.
- [27] COBO Angel, PHP y MySql.tecnologías para el desarrollo de aplicaciones web. Editorial Diaz de Santos. Madrid. 2005, pp 18.
- [28] COBO Angel, PHP y MySql.tecnologías para el desarrollo de aplicaciones web. Editorial Diaz de Santos. Madrid. 2005, pp 23.
- [32] COBO Angel, PHP y MySql.tecnologías para el desarrollo de aplicaciones web. Editorial Diaz de Santos. Madrid. 2005, pp 19.
- [40] ROSA C. M. L., Introducción a los proyectos de ingeniería, 2013.

PROYECTOS DE TITULACIÓN

- [41] AMAYA H. B. M., Desarrollo de un sistema de gestión académica para escuelas primarias Fiscales. EPN. Quito, 2013, pp 39,40.
- [42] CAIZA, J. C. Sistema de solicitudes de órdenes de compra de clientes en un patio de comidas mediante dispositivos móviles en una red inalámbrica. EPN. Quito, 2010, pp 86.
- [43] VISCARRA, D. H. Diseño e implementación de un prototipo de software para helpdesk y distribución de claves de activación de licencias de sistemas vía Web, para la corporación MILENIOSYSTEM S.A. EPN. Quito, 2011, pp 50.
- [44] AMAYA H. B. M., Desarrollo de un sistema de gestión académica para escuelas primarias Fiscales. EPN. Quito, 2013, pp 39,47.
- [45] LLUMIQUINGA, C. V., Desarrollo de un sistema de gestión académica para el COLEGIO NACIONAL RAÚL ANDRADE. EPN. Quito, 2012, pp 89.

PAGINAS WEB

- [2] GARZAS, J. (2013). Metodologías Ágiles. Obtenido de <http://www.javiergarzas.com/metodologias-agiles>
- [3] Wikipedia, "Programación Extrema," [Online]. Available: http://www.es.wikipedia.org/wiki/Programación_extrema.
- [4] LiderDeProyecto.com, «LiderDeProyecto.com,» 2010. [En línea]. Available: http://www.liderdeproyecto.com/articulos/extreme_programming.html.
- [5] H. Tek, «Extreme Programming,» 2010. [En línea]. Available: <http://www.hanantek.com/extremme-programing>.
- [6] Oness, «Ciclo de vida de un proyecto XP,» 2012. [En línea]. Available: <http://oness.sourceforge.net/proyecto/html/ch05s02.html>.
- [7] Wikipedia, «Plataform Technology,» 2014. [En línea]. Available: http://en.wikipedia.org/wiki/Platform_technology.
- [8] J. U., «Tipos de plataformas web,» 2011. [En línea]. Available: <http://www.zainex.es/net/tipos-plataformas-web>.
- [9] Q. Inc., «What is LAMP?,» 2013. [En línea]. Available: <http://www.webopedia.com/TERM/L/LAMP.html>.
- [10] Q. Inc., «What is WAMP?,» 2013. [En línea]. Available: <http://www.webopedia.com/TERM/W/WAMP.html>.
- [11] Q. Inc., «Acronym Guide to Web Stacks,» 2007. [En línea]. Available: http://www.webopedia.com/quick_ref/webstack_acronyms.asp.

- [12] J. U., «Tipos de plataformas web,» 2011. [En línea]. Available: <http://www.zainex.es/net/tipos-plataformas-web>.
- [13] J. U., «Tipos de plataformas web,» 2011. [En línea]. Available: <http://www.zainex.es/net/tipos-plataformas-web>.
- [20] Wikipedia, «Lenguaje de Programación,» 2014. [En línea]. Available: http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n.
- [24] Wikipedia, "Script del lado del servidor," 2010. [Online]. Available: http://es.wikipedia.org/wiki/Script_del_lado_del_servidor.
- [29] Wikipedia, "PHP," 2013. [Online]. Available: <http://es.wikipedia.org/wiki/PHP>.
- [30] L. R. infomática, "Lenguaje de Programación ASP," 2008. [Online]. Available: <http://www.larevistainformatica.com/ASP.htm>.
- [31] Desconocido, "Introducción a JSP," [Online]. Available: <https://sites.google.com/site/conceptoprogramacion/Home/jsp>.
- [33] Wikipedia, "Sistema de gestión de bases de datos," 2014. [Online]. Available: http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos.
- [34] Wikipedia, "Modelo relacional," 2015. [Online]. Available: http://es.wikipedia.org/wiki/Modelo_relacional.
- [35] Wikipedia, "Modelo relacional," 2015. [Online]. Available: http://es.wikipedia.org/wiki/Modelo_relacional.
- [36] Wikipedia, "Sistema de gestión de bases de datos," 2014. [Online]. Available: http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos.
- [37] Wikipedia, "MySQL," 2014. [Online]. Available: <http://es.wikipedia.org/wiki/MySQL>
- [38] R. H. Velasco, "El SGBDR Oracle," [Online]. Available: <http://www2.rhernando.net/modules/tutorials/doc/bd/oracle.html>.
- [39] TECNIE. C. LTDA., "TECNIE OIL ENERGY," 2014. [Online]. Available: www.tecnie.com.
- [46] USABILITY GOV., "SYSTEM USABILITY SCALE," 2007. [Online]. Available: <http://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>

ANEXOS