



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE SITIO *WEB* DE UNIDAD DE EPS

Walter Jacobo Galicia García

Asesorado por la Inga. Floriza Ávila Pesquera de Medinilla

Guatemala, septiembre de 2011

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE SITIO *WEB* DE UNIDAD DE
EPS**

TRABAJO DE GRADUACIÓN

**PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR**

WALTER JACOBO GALICIA GARCÍA

ASESORADO POR LA INGA. FLORIZA ÁVILA PESQUERA DE MEDINILLA

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, SEPTIEMBRE DE 2011

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Inga. Alba Guerrero de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. Juan Carlos Molina Jiménez
VOCAL V	Br. Mario Maldonado Muralles
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADORA	Inga. Floriza Ávila Pesquera de Medinilla
EXAMINADORA	Inga. Sonia Castañeda
EXAMINADOR	Ing. Marlon Pérez Turk
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE SITIO *WEB* DE UNIDAD DE EPS

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha mayo de 2007.



Walter Jacobo Galicia García



Guatemala, 19 de enero de 2011.
REF.EPS.DOC.33.01.2011.

Inga. Norma Ileana Sarmiento Zeceña de Serrano
Directora Unidad de EPS
Facultad de Ingeniería
Presente

Estimada Ingeniera Sarmiento Zeceña.

Por este medio atentamente le informo que como Asesora-Supervisora de la Práctica del Ejercicio Profesional Supervisado, (E.P.S) del estudiante universitario de la Carrera de Ingeniería en Ciencias y Sistemas, **Walter Jacobo Galicia García** Carné No. **9416379** procedí a revisar el informe final, cuyo título es **“ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE SITIO WEB DE UNIDAD DE EPS”**.

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

“Id y Enseñad a Todos”

Inga. Floriza Felipa Avila Pesquera de Medinilla
Asesora-Supervisora de EPS
Área de Ingeniería en Ciencias y Sistemas

FFAPdM/RA





Guatemala, 19 de enero de 2011.
REF.EPS.DOC.36.01.2011.

Ing. Marlon Antonio Pérez Turck
Director Escuela de Ingeniería Ciencias y Sistemas
Facultad de Ingeniería
Presente

Estimado Ingeniero Perez Turck.

Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **“ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE SITIO WEB DE UNIDAD DE EPS”**, que fue desarrollado por el estudiante universitario **Walter Jacobo Galicia García** Carné No. **9416379** quien fue debidamente asesorado por el Ing. Jorge Armin Mazariegos y supervisado por la Inga. Floriza Felipa Ávila Pesquera de Medinilla.

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte del Asesor y la Supervisora de EPS, en mi calidad de Directora apruebo su contenido solicitándole darle el trámite respectivo.

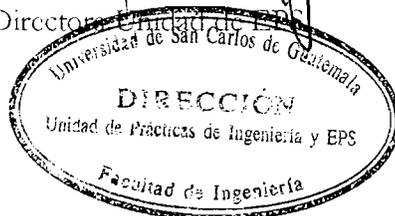
Sin otro particular, me es grato suscribirme.

Atentamente,

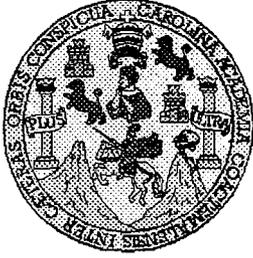
“Id y Enseñad a Todos”

Inga. Norma Ileana Sarmiento Zeceña de Serrano

Directora Unidad de EPS



NISZ/ra



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 16 de Febrero de 2011

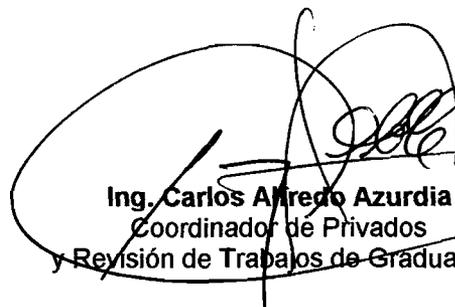
Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación-EPS del estudiante **WALTER JACOBO GALICIA GARCIA**, carné **1994-16379**, titulado: **“ANALISIS, DISEÑO E IMPLEMENTACION DE SITIO WEB DE UNIDAD EPS”**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



E
S
C
U
E
L
A

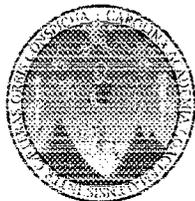
D
E

C
I
E
N
C
I
A
S

Y

S
I
S
T
E
M
A
S

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

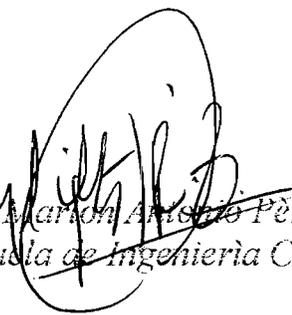


FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, de trabajo de graduación titulado **“ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE SITIO WEB DE UNIDAD DE EPS”**, presentado por el estudiante **WALTER JACOBO GALICIA GARCÍA**, aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”

Ing. Marlon Anselio Pérez Tur
Director, Escuela de Ingeniería Ciencias y Sistemas



Guatemala, 01 de septiembre 2011



DTG. 325.2010

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE SITIO WEB DE UNIDAD DE EPS**, presentado por el estudiante universitario **Walter Jacobo Galicia García**, autoriza la impresión del mismo.

IMPRÍMASE:

Ing. Murphy Olimpo Paiz Recinos
Decano



Guatemala, 5 de septiembre de 2011.

/gdech

ACTO QUE DEDICO A:

- Dios** Por darme la vida, ser mi guía y mi fortaleza en los momentos difíciles que se presentaron, porque sin él no hubiese alcanzado la meta.
- Mis padres** Carlos Leónidas Galicia Ochoa y Martha Rubenia García Ayala, por darme la oportunidad de estar aquí y apoyarme en todo momento en este difícil trayecto, porque todo sacrificio tiene su recompensa y esta recompensa les pertenece a ellos.
- Mis hermanas** Mayra Esther y Mauda Raquel por brindarme su apoyo incondicional y estar en los momentos precisos de sus apoyos.
- Mis amigos** Por brindarme su amistad, apoyo y compañerismo, así como a mis amigos de carrera, por tantos momentos y desvelos compartidos durante los proyectos que realizamos.
- Mis sobrinos** Por su cariño hacia mi persona.
- Mi compañera de hogar** Maritza Portillo, por ser esa persona quien me acompaña cada día por el camino de la vida y que cada momento me ha brindado su amor y apoyo incondicional.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
GLOSARIO	IX
RESUMEN	XVII
OBJETIVOS	XIX
INTRODUCCIÓN	XXI
1. FASE DE INVESTIGACIÓN	1
1.1. Arquitectura de un proyecto <i>web</i>	1
1.1.1. Aplicación <i>web</i>	1
1.1.2. Interfaz	2
1.1.3. Consideraciones técnicas	2
1.1.4. Estructura de una aplicación <i>web</i>	3
1.2. Arquitectura <i>web</i>	4
1.2.1. Elementos de la arquitectura cliente/servidor	4
1.2.2. Características del modelo cliente/servidor	6
2. FASE TÉCNICO PROFESIONAL	9
2.1. Arquitectura de aplicaciones <i>.net</i>	9
2.1.1. Objetivo principal del diseño de aplicaciones distribuidas	9
2.1.2. Servicios e integración de servicios	10
2.1.3. Componentes y niveles en aplicaciones y servicios	11
2.2. Directivas de seguridad, administración operativa y comunicaciones	12
2.2.1. Tipos de componentes	12
2.2.2. Diseño de capas de presentación	14

2.2.3.	Diseño de capas empresariales	15
2.2.4.	Diseño de capa de datos	16
2.3.	Diseño de aplicaciones y servicios	17
2.3.1.	Diseño de la directiva de seguridad	17
2.3.1.1.	Principios generales sobre seguridad	18
2.3.1.2.	Autenticación	19
2.3.1.3.	Flujo de identidad entre los niveles	20
2.3.1.4.	Autorización	20
2.3.2.	Diseño de la directiva de administración operativa	21
2.3.2.1.	Administración de excepciones	22
2.3.2.2.	Ubicación de servicios	22
2.3.3.	Diseño de la directiva de comunicaciones	22
2.4.	Implementación física y requisitos operativos	23
2.4.1.	Implementación de los componentes de la aplicación	23
2.4.2.	Entornos físicos de implementación	24
2.4.3.	Requisitos operativos	25
3.	FASE DE ENSEÑANZA APRENDISAJE	27
3.1.	Justificaciones del sitio <i>web</i>	27
3.1.1.	Justificación general de la aplicación	27
3.1.2.	Justificación técnica	27
3.1.3.	Justificación económica	28
3.1.4.	Justificación social	28
3.2.	Tecnología utilizada	29
3.2.1.	Tecnología de desarrollo	29
3.2.2.	Tecnología de edición de texto	31
3.2.2.1.	Editor de texto <i>HTML WYSIWYG</i> de <i>TINYMCE</i>	31
3.3.	Recursos disponibles	33
3.3.1.	Recurso Humano	33

3.3.2.	Recursos de materiales	34
3.3.3.	Recurso financiero	34
3.3.4.	Presupuesto	34
3.4.	Diagramas del proyecto	36
3.4.1.	Diagrama entidad relación	36
3.4.2.	Diagramas de red	37
3.4.2.1.	Diagrama de red 1	37
3.4.2.2.	Diagrama de red 2	38
4.	IMPLEMENTACIÓN DEL PROYECTO <i>WEB</i> EPS	39
4.1.	Guías de implementación	39
4.1.1.	Guía técnica	39
4.1.1.1.	Guía para la instalación del servidor <i>web Internet information server</i> sobre <i>Windows 2003</i>	39
4.1.1.2.	Guía para la configuración del servidor en <i>trusted source</i>	41
4.1.2.	Guía administrativa	45
4.1.2.1.	Para ingresar a la <i>web</i> de administración	45
4.1.2.2.	Modificar datos de director	46
4.1.2.3.	Modificar datos de personal docente	47
4.1.2.4.	Modificar directores antiguos	48
4.1.2.5.	Modificar información de organigrama	50
4.1.2.6.	Modificar descripción departamento EPS	52
4.1.2.7.	Modificar visión y misión EPS	52
4.1.2.8.	Modificar reseña histórica	53
4.1.2.9.	Subir normativo de EPS	54
4.1.2.10.	Subir normativo de prácticas	56
4.1.2.11.	Modificar horario atención de departamento EPS	57
4.1.2.12.	Modificar horario atención EPS	58

4.1.2.13. Modificar horario atención prácticas iniciales	59
4.1.2.14. Modificar horario atención prácticas intermedias	59
4.1.2.15. Modificar horario atención prácticas finales	60
4.1.2.16. Subir horario de clases prácticas iniciales	61
4.1.2.17. Subir horario de clases prácticas intermedias	63
4.1.2.18. Ingreso de imágenes	64
4.1.2.19. Ingreso de avisos	67
4.1.2.20. Ingreso de talleres	69
4.1.2.21. Cambiar la contraseña de usuario administrador	71
CONCLUSIONES	73
RECOMENDACIONES	75
BIBLIOGRAFIA	77

ÍNDICE DE ILUSTRACIONES

FIGURAS

1. Arquitectura en tres capas	4
2. Aplicaciones cliente/servidor	5
3. Arquitectura cliente/servidor	6
4. Características del modelo cliente/servidor	6
5. Proceso implementado utilizando servicios	11
6. Solución basada en servicios	12
7. Tipos de componentes	13
8. Aspectos de la directiva de seguridad	17
9. Aspectos de la directiva de administración operativa	21
10. Aspectos de la directiva de comunicaciones	22
11. Arquitectura de <i>Internet datacenter</i>	24
12. Diagrama entidad relación	36
13. Diagrama de red 1	37
14. Diagrama de red 2	38
15. Agregar o quitar programas	39
16. Agregar o quitar componentes de <i>Windows</i>	40
17. Asistente para componentes de <i>Windows</i>	40
18. <i>SQL Enterprise Manager</i>	41
19. Propiedades de logueo de SQL	42
20. Roles de base de datos	42
21. Acceso a base de datos	43
22. Administrador de <i>Internet Information Server</i>	43
23. Propiedades de proyecto EPS	44

24.	Métodos de autenticación	44
25.	Área administrativa EPS	45
26.	Página de logueo	46
27.	Modificar datos director	46
28.	Modificación de datos personal	47
29.	Editar personal docente	47
30.	Agregar nuevo personal	48
31.	Modificar directores antiguos	49
32.	Edición de datos de directores	49
33.	Agregar datos de nuevo director	50
34.	Modificar información de organigrama	51
35.	Modificando organigrama de EPS	51
36.	Modificar descripción de departamento	52
37.	Modificar información de misión y visión	53
38.	Modificar reseña histórica	54
39.	Subir normativo de EPS	55
40.	Seleccionando y subiendo normativo de EPS	55
41.	Subir normativo de prácticas de ingeniería	56
42.	Seleccionando y subiendo normativo de prácticas	57
43.	Modificar horario de atención unidad de EPS	58
44.	Modificar horario de atención de EPS	58
45.	Modificar horario de atención de prácticas iniciales	59
46.	Modificar horario de atención de prácticas intermedias	60
47.	Modificar horario de atención de prácticas finales	61
48.	Subir horario de cursos de prácticas iniciales	62
49.	Seleccionando y subiendo horario de atención de prácticas iniciales	62
50.	Subir horario de clases de prácticas intermedias	63
51.	Seleccionando y subiendo horario de clases de prácticas intermedias	64
52.	Subir imágenes	65

53.	Subir detalles de imagen 1	65
54.	Subir detalles de imagen 2	66
55.	Seleccionando formato y posición de imagen a subir	66
56.	Vista previa de imagen	67
57.	Subir avisos de EPS	67
58.	Subir detalles de aviso de EPS 1	68
59.	Subir detalles de aviso de EPS 2	68
60.	Subir detalles de aviso de EPS 3	69
61.	Subir talleres	70
62.	Subir detalles de talleres 1	70
63.	Subir detalles de talleres 2	71
64.	Cambiar contraseña de administrador	71

TABLAS

I.	Presupuesto	35
----	-------------	----

GLOSARIO

<i>Applets</i>	Componentes escritos en lenguaje <i>Java</i> los cuales pueden incorporarse en un sitio <i>web</i> brindándole nueva funcionalidad.
Base de Datos	Conjunto de datos que pertenecen al mismo contexto almacenado sistemáticamente para su uso posterior.
Capa	Una capa se puede concebir como un patrón de arquitectura en el que los componentes utilizan servicios en las capas inferiores. La utilización de capas facilita el mantenimiento. La comunicación entre dos capas determina la facilidad con que se podrá particionar la aplicación en ese punto para la distribución física a través de los niveles. Unos esquemas de capas estrictos no permiten a las capas tener acceso a otras capas que no sean las inmediatamente inferiores, mientras que unos esquemas de capas más flexibles permiten a una capa determinada utilizar cualquier otra que esté por debajo de ella.
<i>CGI(Common Gateway Interface)</i>	Interfaz de entrada común, se refiere al uso de la consola del sistema operativo para presentar y procesar datos desde las páginas <i>HTML</i> y hacerlas más dinámicas.
Componente	Es una parte de un sistema. Es una unidad de funcionalidad que se puede amortizar a través de diversas implementaciones. Un componente generalmente se implementa como un objeto de <i>software</i> que expone varias interfaces con lógica.

Concentrador(<i>HUB</i>)	Dispositivo que permite centralizar el cableado de una red, también conocido con el nombre de <i>hub</i> .
Conmutividad	Es un patrón de diseño para una implementación en la que los mensajes tendrán el mismo resultado.
Consola	Interface de comandos de un sistema operativo que permite el envío de órdenes a la computadora a través del teclado.
Correo Electrónico	Es un servicio brindado en una red que permite a los usuarios el envío y recepción de mensajes en un medio electrónico.
Corta fuegos (<i>Firewall</i>)	Equipo de <i>hardware</i> o <i>software</i> utilizado en las redes de computadoras para prevenir y proteger un sitio privado conectado a <i>Internet</i> o cualquier otra red ajena a la propia, mediante políticas de comunicación, acceso y restricciones de la red, según las necesidades de los usuarios.
Cuadro de diálogo	Ventanas que muestran un mensaje donde se solicita una respuesta del usuario para una determinada situación.
Dirección <i>IP</i>	Es la identificación de una máquina en concreto dentro de la red <i>TCP/IP</i> a la que pertenece.
Dirección <i>URL</i> (<i>Uniform Resource Locator</i>)	Dirección de dominio o conjunto de caracteres alfanuméricos con los que se identifica de manera unívoca un determinado sitio <i>web</i> de <i>Internet</i> .

<i>DNS (Domain Name System)</i>	Conjunto de protocolos y servicios que permite a los usuarios utilizar nombres en lugar de direcciones <i>IP</i> numéricas.
Enlace (<i>Link</i>)	Vínculo dentro de una página hacia otra página que permite realizar una acción o re direccionar el buscador a otra página.
Ensamblado	Un ensamblado es una unidad de implementación en una aplicación basada en <i>.net framework</i> .
<i>Filter Transfer Protocol (FTP)</i>	Protocolo o conjunto de reglas que permiten la transferencia de archivos dentro de una red de computadoras.
Flujo de trabajo	Proceso empresarial en el que los pasos se deben realizar en un determinado orden, y se deben cumplir unas condiciones predefinidas, antes de avanzar de un paso al siguiente.
Formulario <i>web</i>	Página <i>web</i> interactiva con secciones predefinidas para llenar u opciones para seleccionar. En ella se recolectan datos para tratarlos, generalmente mediante un <i>script CGI</i> en el servidor.
<i>Hardware</i>	Dispositivos físicos que comprenden un sistema de computación. Su complemento es el <i>software</i> .
Herramienta de <i>software</i>	Aplicaciones de <i>software</i> de uso sencillo, creadas para ayudar en la realización de tareas específicas.
<i>HTML</i>	Lenguaje de etiquetas de hipertexto. Conjunto de códigos y expresiones simbólicas insertas en un archivo para la correcta interpretación de una página de <i>Internet</i> . El hipertexto hace

referencia a la capacidad de navegación. Las etiquetas hacen referencia a las instrucciones para realizar una acción.

Interfaz Una interfaz es la parte de un programa informático que permite a éste comunicarse con el usuario o con otras aplicaciones permitiendo el flujo de información.

Interfaz de servicios Es un punto de entrada para un servicio. Proporciona una interfaz pública que los llamadores pueden utilizar para consultar el contrato que admite la interfaz y realizar llamadas de método basado en mensajes al servicio.

Internet Red de redes a escala mundial de millones de computadoras interconectadas con el conjunto de protocolos *TCP/IP*.

Intranet Una *intranet* es una red local que utiliza herramientas de *Internet*. Se puede considerar como un *Internet* privado que funciona dentro de una organización. Normalmente, dicha red local tiene como base el protocolo *TCP/IP* de *Internet* y utiliza un sistema *firewall* (cortafuegos) que no permite acceder a la misma desde el exterior.

Lenguaje de programación Serie de instrucciones que son ejecutadas en secuencia construidas con base en una sintaxis y una gramática especial.

Licencia Documento que autoriza, a quien lo posee, la práctica de una actividad.

Metodología Parte de la lógica que estudia los métodos. Se divide en dos

partes: sistemática, fija las normas de la definición, de la división, de la clasificación y de la prueba, e inventiva, fija las normas de los métodos de investigación propios de cada ciencia.

Multiplataforma	Se refiere a una aplicación de <i>software</i> que puede funcionar en diferentes sistemas operativos o diferentes ambientes de trabajo.
Navegador de Internet	Aplicación de <i>software</i> utilizada para desplegar el contenido de las páginas y desplazarse entre los sitios de <i>Internet</i> .
Página principal	Conocida como <i>homepage</i> o portada. Contiene enlaces a partir de los cuales se pueden visitar contenidos de un sitio en <i>web</i> .
Palabra reservada	Palabra que no puede usarse para propósitos distintos de los establecidos por el programa en uso.
PDF	<i>Portable document format</i> ó formato de despliegue de documentos que permite la visualización del archivo tal y como fue creado, no es modificable y reduce el espacio de almacenamiento.
Procesador	Conjunto de circuitos lógicos que procesa las instrucciones básicas de una computadora.
Propiedad intelectual	Derechos relativos a las creaciones y actividades intelectuales en los terrenos industrial, científico, literario y artístico.
Protocolo TCP/IP	Conjunto de protocolos de red que implementa la pila de protocolos en la que se basa <i>Internet</i> y que permiten la

transmisión de datos entre redes de computadoras. También se la denomina conjunto de protocolos *TCP/IP*, en referencia a los dos protocolos más importantes que la componen: protocolo de control de transmisión (*TCP*) y protocolo de *Internet (IP)*⁷.

Prototipo Modelo de prueba de un proyecto en una de sus fases de diseño o desarrollo que aun no es el final.

Red local Conjunto de computadoras conectadas mediante dispositivos de comunicación que se encuentran en un mismo edificio.

Scripts Un conjunto de comandos escritos en un lenguaje interpretado para automatizar ciertas tareas de aplicación.

Segmentación Proceso de dividir un grupo mayor de objetos en grupos más pequeños que tengas características semejantes.

Servicio Componente de *software* que se puede utilizar en una parte de un proceso empresarial completo. Los servicios admiten interfaz de comunicación basada en mensajes, a través de la cual tiene lugar una conversación. Un servicio encapsula su propio estado y datos empresariales, y la comunicación con él únicamente se puede realizar a través de las interfaces de servicio que expone.

Servidor Computadora, que distribuye los recursos dentro de una red proveyendo la información requerida por los usuarios.

Sistema manejador de bases de datos (DBMS) Una base de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su uso

posterior. Conjunto organizado e integrado de datos almacenados en computadora, con el fin de facilitar su uso para aplicaciones con múltiples finalidades.

Sitio web Conjunto de páginas *web* accesibles desde *Internet* mediante un navegador, el sitio es identificado con una dirección (*URL*).

Software A diferencia del *hardware*, es lo que no se ve, es decir los programas y aplicaciones que están guardadas en un disco duro, *CD-ROM* o *disquetes*.

Tecnología SSL Tecnología que utiliza el protocolo de seguridad en la capa de conexión a una red, es decir un conjunto de reglas y procesos establecidos que permite una conexión segura.

Transacción Operación que modifica el estado de una base de datos, sin que los datos en sí mismos pierdan consistencia alguna, una transacción debe ser equivalente a una interacción atómica, es decir, que se realice de una sola vez.

Virtual Que no tiene existencia física material y tangible, sino conceptual y abstracta.

Web master Es un término comúnmente usado que se refiere a la o las personas responsables de administrar un sitio *web* específico.

RESUMEN

En el actual trabajo de graduación se presenta un sistema de información para el estudiante que ayudará a complementar la cooperación entre diferentes profesionales dispuestos a compartir su trabajo y conocimiento.

El presente trabajo consiste en una estrategia de participación que permita al estudiante la posibilidad de contar con un mejor enfoque de hacia dónde se dirige la tecnología, buscando la colaboración por parte de profesionales, catedráticos que a través de publicaciones aborden temas que son útiles dentro de los cursos del área así como para estudiantes que estén realizando EPS, y así garantizar la calidad de la educación y elevar el grado de conocimientos diversos por parte de los estudiantes de las áreas profesionales.

Actualmente la tecnología *web* tiene un impacto dado por su carácter innovador, multiplicador, integracionista y de perfeccionamiento académico. Existen medios modernos de comunicación virtual, pero los mismos no han sustituido, sino más bien complementado, el potencial de la vinculación directa y presencial entre docentes y alumnos.

OBJETIVOS

General

Diseñar e implementar el sitio *web* de la unidad de EPS que mejor se adapte a sus funciones y necesidades, para llevar los servicios prestados a un nivel más alto con calidad y exactitud de información.

Específicos

1. Tener presencia como sitio *web* de EPS.
2. Reducir el problema de información y accesibilidad a EPS.
3. Mejorar el desempeño y rendimiento del sitio *web* de EPS.
4. Diseñar un sistema de base de datos capaz de almacenar toda la información del sitio.
5. Analizar, diseñar e implementar el sitio *web* de la Unidad de EPS.
6. Publicar electrónicamente el sitio *web*, de forma que pueda llegar a la mayoría de la población estudiantil.

INTRODUCCIÓN

El incesante avance del conocimiento genera la necesidad de que las instituciones de educación superior promuevan estrategias que permitan, por un lado, mantener su nivel académico adecuado en cada una de sus áreas de estudio y por otro, cubrir las nuevas y emergentes tecnologías o líneas de investigación, además de mantener la información al día de todo lo que cada departamento cubre. En la actualidad, es casi imposible pensar en una universidad, por grande o completa que sea, que pueda abarcar en su totalidad el enorme número de áreas diversas de conocimiento existentes. Una de las maneras más efectiva y eficiente es mantener a todo el alumnado informado, con respecto a cada una de las actividades que se realizan así como tener un medio a través del cual se pueda mantener informado al docente o catedrático así como a las personas que realizan sus actividades finales de su carrera.

Es así como se plantea un mecanismo de información al estudiante para complementar la cooperación entre diferentes profesionales dispuestos a compartir su trabajo y conocimiento.

El sitio *web* de la unidad de EPS de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, consiste en una estrategia de participación que con una serie de accesos les permita a los estudiantes la posibilidad de contar con un mejor enfoque de hacia donde se dirige la tecnología, buscando la colaboración por parte de profesionales, catedráticos que a través de publicaciones aborden temas que son útiles dentro de los cursos del área así como para estudiantes que estén realizando EPS, y así garantizar la calidad de la educación y elevar el grado de conocimientos diversos por parte de los estudiantes de las áreas profesionales.

El impacto *web* está dado por su carácter innovador, multiplicador, integracionista y de perfeccionamiento académico. Cabe señalar asimismo, que si bien existen medios modernos de comunicación virtual, ellos no han sustituido, sino más bien complementado, el potencial de la vinculación directa y presencial entre docentes y alumnos.

Es así como se presenta esta *web* que desea cubrir cada una de las partes que integran el Depto. de EPS, y ser un ente informativo para el estudiante a través de cada una de las partes que lo conforman así como acelerar los procedimientos realizados actualmente en el departamento.

1. FASE DE INVESTIGACIÓN

1.1. Arquitectura de un Proyecto *web*

El describir la arquitectura de un proyecto *web* es una prioridad de todo trabajo relacionado, así también el describir los conceptos técnicos que son utilizados para desarrollarlo. Las fases de diseño y desarrollo de la aplicación son las ideales para determinar las necesidades de seguridad y construcción de un sitio *web*.

1.1.1. Aplicación *web*

En la ingeniería *software* se denomina aplicación *web* a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor *web* a través de *Internet* o de una *intranet* mediante un navegador. En otras palabras, es una aplicación *software* que se codifica en un lenguaje soportado por los navegadores *web* (*HTML*, *JavaScript*, *Java*, *asp.net*, etc.) en la que se confía la ejecución al navegador.

Las aplicaciones *web* son populares debido a lo práctico del navegador *web* como cliente ligero, así como a la facilidad para actualizar y mantener aplicaciones *web* sin distribuir e instalar *software* a miles de usuarios potenciales. Existen aplicaciones como los *webmails*, *wikis*, *weblogs*, tiendas en línea y la propia *Wikipedia* que son ejemplos bien conocidos de aplicaciones *web*.

Es importante mencionar que una página *web* puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a

cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

1.1.2. Interfaz

Las interfaces *web* tienen ciertas limitaciones en las funcionalidades que se ofrecen al usuario. Hay funcionalidades comunes en las aplicaciones de escritorio como dibujar en la pantalla o arrastrar-y-soltar que no están soportadas por las tecnologías *web* estándar. Los desarrolladores *web* generalmente utilizan lenguajes interpretados o *script* en el lado del cliente para añadir más funcionalidades, especialmente para ofrecer una experiencia interactiva que no requiera recargar la página cada vez (lo que suele resultar molesto a los usuarios). Recientemente se han desarrollado tecnologías para coordinar estos lenguajes con tecnologías en el lado del servidor, como por ejemplo *PHP*. Como ejemplo, *ajax*, es una técnica de desarrollo *web* que usa una combinación de varias tecnologías.

1.1.3. Consideraciones técnicas

Una ventaja significativa es que las aplicaciones *web* deberían funcionar igual independientemente de la versión del sistema operativo instalado en el cliente. En lugar de crear clientes para *Windows*, *Mac OS X*, *GNU/Linux*, y otros sistemas operativos, la aplicación *web* se escribe una vez y se ejecuta igual en todas partes. Sin embargo, hay aplicaciones inconsistentes escritas con *HTML*, *CSS*, *DOM* y otras especificaciones para navegadores *web* que pueden causar problemas en el desarrollo y soporte de las aplicaciones *web*. Adicionalmente, la posibilidad de los usuarios de personalizar muchas de las características de la interfaz (tamaño y color de fuentes, tipos de fuentes, inhabilitar *Javascript*) puede interferir con la consistencia de la aplicación *web*.

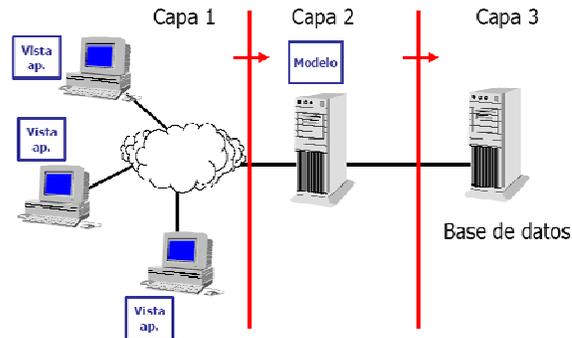
Otra aproximación es utilizar *Adobe Flash Player* o *Java applets* para desarrollar parte o toda la interfaz de usuario. Como casi todos los navegadores incluyen soporte para estas tecnologías (usualmente por medio de *plug-ins*), las aplicaciones basadas en *Flash* o *Java* pueden ser implementadas con aproximadamente la misma facilidad. Dado que ignoran las configuraciones de los navegadores, estas tecnologías permiten más control sobre la interfaz, aunque las incompatibilidades entre implementaciones *Flash* o *Java* puedan crear nuevas complicaciones. Por las similitudes con una arquitectura cliente-servidor, con un cliente "no ligero", existen discrepancias sobre el hecho de llamar a estos sistemas "aplicaciones *web*"; un término alternativo es "aplicación enriquecida de *Internet*".

1.1.4. Estructura de una aplicación *web*

La estrategia tradicional de utilizar aplicaciones compactas causa gran cantidad de problemas de integración en sistemas *software* complejos como pueden ser los sistemas de gestión de una empresa o los sistemas de información integrados consistentes en más de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad, integración.

Para solventar estos problemas se ha generalizado la división de las aplicaciones en capas que normalmente serán tres: una capa que servirá para guardar los datos (base de datos), una capa para centralizar la lógica de negocio (modelo) y por último una interfaz gráfica que facilite al usuario el uso del sistema.

Figura1. **Arquitectura en Tres Capas**



Fuente: Carlos Sánchez, ONess: un proyecto *open source* para el negocio textil mayorista desarrollado con tecnologías *open source* innovadoras.
<http://oness.sourceforge.net/proyecto/html/index.html>.

1.2. **Arquitectura web**

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficazmente dentro de la organización. Se debe señalar que para seleccionar el modelo de una arquitectura, hay que partir del contexto tecnológico y organizativo del momento y, que la arquitectura cliente/servidor requiere una determinada especialización de cada uno de los diferentes componentes que la integran.

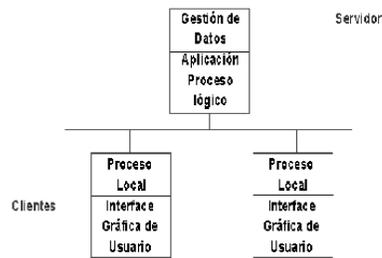
1.2.1. **Elementos de la arquitectura cliente/servidor**

En esta aproximación, y con el objetivo de definir y delimitar el modelo de referencia de una arquitectura cliente/servidor, se debe identificar los componentes que permitan articular dicha arquitectura, considerando que toda aplicación de un sistema de información está caracterizada por tres componentes básicos:

- a) Presentación/captación de información
- b) Procesos
- c) Almacenamiento de la información

Los cuales se suelen distribuir tal como se presenta en la figura 2

Figura 2. **Aplicaciones cliente/servidor**



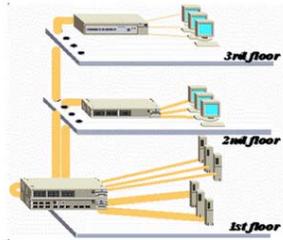
Fuente: Instituto Nacional de Estadística e Informática, Tecnología cliente/servidor.
<http://www.inei.gob.pe/biblioineipub/bancopub/inf/Lib5038/indice.HTM>.

Y se integran en una arquitectura cliente/servidor en base a los elementos que caracterizan dicha arquitectura, es decir:

- a) Puestos de trabajo
- b) Comunicaciones
- c) Servidores

Tal como se presenta en la figura 3

Figura3. **Arquitectura cliente/servidor**



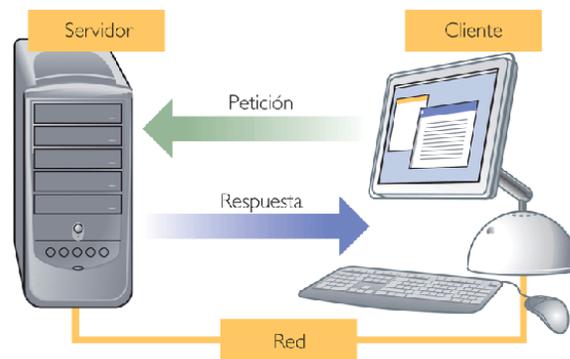
Fuente: Wilson Ayala, Arquitectura cliente/servidor. <http://wilsonayala.blogspot.com>.

1.2.2. Características del modelo cliente/servidor

En el modelo cliente/servidor se puede encontrar las siguientes características:

- a) El cliente y el servidor puede actuar como una sola entidad y también puede actuar como entidades separadas, realizando actividades o tareas independientes.
- b) Las funciones de cliente y servidor pueden estar en plataformas separadas, o en la misma plataforma.

Figura 4. **Características del modelo cliente/servidor**



Fuente: Paola y Juan, Clasificación de las redes.

http://wwwpayojohn.blogspot.com/2010_05_01_archive.html.

- c) Un servidor da servicio a múltiples clientes en forma concurrente.
- d) Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los clientes o de los servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- e) La interrelación entre el *hardware* y el *software* están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
- f) Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final. También es importante hacer notar que las funciones cliente/servidor pueden ser dinámicas.
- g) Además se constituye como el nexo de unión más adecuado para reconciliar los sistemas de información basados en *mainframes* o minicomputadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo.
- h) Designa un modelo de construcción de sistemas informáticos de carácter distribuido. Su representación típica es un centro de trabajo (*PC*), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la

organización, al tiempo que puede acceder a los recursos de este *host* central y otros sistemas de la organización ponen a su servicio.

En conclusión, cliente/servidor puede incluir múltiples plataformas, bases de datos, redes y sistemas operativos. Estos pueden ser de distintos proveedores, en arquitecturas propietarias y no propietarias y funcionando todos al mismo tiempo. Por lo tanto, su implantación involucra diferentes tipos de estándares: *APPC*, *TCP/IP*, *OSI*, *NFS*, *DRDA* corriendo sobre *DOS*, *OS/2*, *Windows* o *PC UNIX*, en *TokenRing*, *Ethernet*, *FDDI* o medio coaxial, sólo por mencionar algunas de las posibilidades.

2. FASE TÉCNICO PROFESIONAL

2.1. Arquitectura de aplicaciones .net

2.1.1. Objetivo principal del diseño de aplicaciones distribuidas

El diseño de una aplicación distribuida implica la toma de decisiones sobre su arquitectura lógica y física, así como sobre la tecnología e infraestructura que se emplearán para implementar su funcionalidad. Para la toma de estas decisiones, se debe tener un conocimiento claro de los procesos empresariales que realizará la aplicación (sus requisitos funcionales), así como los niveles de escalabilidad, disponibilidad, seguridad y mantenimiento necesarios (sus requisitos no funcionales, funcionales u operativos).

El objetivo consiste en diseñar una aplicación con las siguientes características:

- a) Solucione el problema empresarial para el que se diseña.
- b) Se tenga en consideración la seguridad desde el principio, teniendo en cuenta los mecanismos adecuados de autenticación, la lógica de autorización y la comunicación segura.
- c) Proporcione un alto rendimiento y esté optimizada para operaciones frecuentes entre patrones de implementación.
- d) Esté disponible y sea resistente, capaz de implementarse en centros de datos de alta disponibilidad y redundantes.

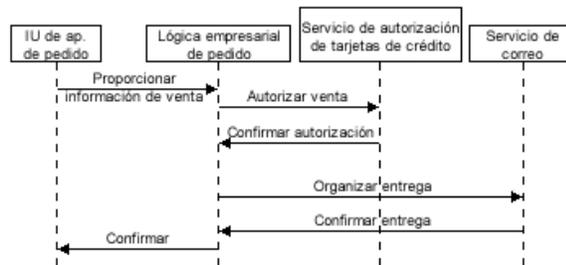
- e) Permita la escalabilidad para cumplir las expectativas de la demanda y admita un gran número de actividades y usuarios con el mínimo uso de recursos.
- f) Se pueda administrar, permitiendo a los operadores implementar, supervisar y resolver los problemas de la aplicación en función del escenario.
- g) Se pueda mantener. Cada parte de funcionalidad debería tener una ubicación y diseño predecibles teniendo en cuenta distintos tamaños de aplicaciones, equipos con habilidades variadas y requisitos técnicos y cambios empresariales.
- h) Funcione en distintos escenarios de aplicaciones y patrones de implementación.

2.1.2. Servicios e integración de servicios

A medida que crece *Internet* y las tecnologías relacionadas, y las organizaciones buscan integrar sus sistemas entre límites de departamentos y de organización, ha evolucionado un enfoque de generación de soluciones basado en servicios. Desde el punto de vista del consumidor, los servicios son conceptualmente similares a los componentes tradicionales, salvo que los servicios encapsulan sus propios datos y no forman parte de la aplicación sino que son utilizados por ésta. Aplicaciones y servicios que necesitan integrarse se pueden generar en distintas plataformas, por distintos equipos, en diferentes programas y se pueden mantener y actualizar independientemente.

El término servicio se utiliza en esta guía para hacer referencia a los componentes de *software* externos que proporcionan servicios empresariales. Esto incluye, aunque no exclusivamente, los servicios *web XML*. Los servicios exponen una interfaz de servicios a la que se envían todos los mensajes entrantes. La definición del conjunto de mensajes que se deben intercambiar con un servicio para que éste realice una tarea empresarial específica es un contrato.

Figura 5. **Proceso implementado utilizando servicios**



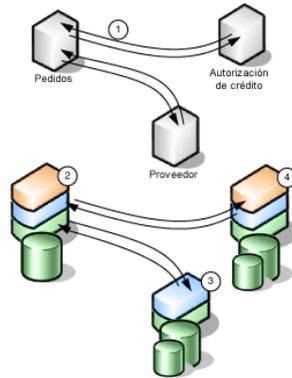
Fuente: *Microsoft Corporation*, Diseño de aplicaciones y servicios. <http://msdn.microsoft.com/es-es/library/ms978340.aspx>.

2.1.3. Componentes y niveles en aplicaciones y servicios

Se ha convertido en un principio ampliamente aceptado en el diseño de aplicaciones distribuidas la división de la aplicación en componentes que ofrezcan servicios de presentación, empresariales y de datos. Los componentes que realizan tipos de funciones similares se pueden agrupar en capas, que en muchos casos están organizados en forma de apilamiento para que los componentes que se encuentran por "encima" de una capa determinada utilicen los servicios proporcionados por ésta, y un componente específico utilizará la funcionalidad proporcionada por otros componentes de su propia capa, y otras capas "inferiores", para realizar su trabajo.

Se utiliza el término capa para hacer referencia a un tipo de componente y el término nivel para hacer referencia a los patrones de distribución físicos. Se puede considerar que la solución basada en servicios está formada por varios servicios, los cuales se comunican entre sí pasando mensajes. Desde el punto de vista conceptual, los servicios se pueden considerar como componentes de la solución global. Sin embargo, internamente el servicio está formado por componentes de *software*, al igual que cualquier otra aplicación, los cuales se pueden agrupar de forma lógica en servicios de presentación, empresariales y de datos, tal y como se muestra en la figura 6.

Figura 6. **Solución basada en servicios**



Fuente: *Microsoft Corporation*, Diseño de aplicaciones y servicios. <http://msdn.microsoft.com/es-es/library/ms978340.aspx>.

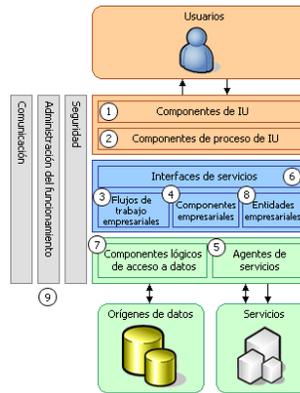
2.2. Directivas de seguridad, administración operativa y comunicaciones

2.2.1. Tipos de componentes

El análisis de la mayoría de las soluciones basadas en modelos de componentes por capas muestra que existen varios tipos de componentes habituales.

Aunque la lista que se muestra en la figura 7 no es completa, representa los tipos de componentes de *software* más comunes encontrados en la mayoría de las soluciones distribuidas.

Figura 7. Tipos de componentes



Fuente: *Microsoft Corporation*, Diseño de aplicaciones y servicios. <http://msdn.microsoft.com/es-es/library/ms978340.aspx>.

Los tipos de componentes identificados en el escenario de diseño son:

- Componentes de interfaz de usuario (IU). La mayor parte de las soluciones necesitan ofrecer al usuario un modo de interactuar con la aplicación.
- Componentes de proceso de usuario. En un gran número de casos, la interacción del usuario con el sistema se realiza de acuerdo a un proceso predecible.
- Flujos de trabajo empresariales. Una vez que el proceso de usuario ha recopilado los datos necesarios, éstos se puede utilizar para realizar un proceso empresarial. Gran parte de los procesos empresariales con llevan la realización de varios pasos, los cuales se deben organizar y llevar a cabo en un orden determinado.
- Componentes empresariales. Independientemente de si el proceso empresarial consta de un único paso o de un flujo de trabajo organizado, la aplicación requerirá el uso de componentes que implementen reglas empresariales y realicen tareas

empresariales. Los componentes empresariales implementan la lógica empresarial de la aplicación.

- e) Agentes de servicios. Cuando un componente empresarial requiere el uso de la funcionalidad proporcionada por un servicio externo, tal vez sea necesario hacer uso de código para administrar la semántica de la comunicación con dicho servicio.
- f) Interfaces de servicios. Para exponer lógica empresarial como un servicio, es necesario crear interfaces de servicios que admitan los contratos de comunicación que requieren los clientes.
- g) Componentes lógicos de acceso a datos. La mayoría de las aplicaciones y servicios necesitan obtener acceso a un almacén de datos en un momento determinado del proceso empresarial.
- h) Componentes de entidad empresarial. La mayoría de las aplicaciones requieren el paso de datos entre distintos componentes.
- i) Componentes de seguridad, administración operativa y comunicación. La aplicación probablemente utilice también componentes para realizar la administración de excepciones, autorizar a los usuarios a que realicen tareas determinadas y comunicarse con otros servicios y aplicaciones.

2.2.2. Diseño de capas de presentación

La capa de presentación contiene los componentes necesarios para habilitar la interacción del usuario con la aplicación. Las capas de presentación más simples contienen componentes de interfaz, como formularios de *Windows Forms* o formularios

web de *asp.net*. Las inter actuaciones más complejas con llevan el diseño de componentes de proceso de usuario que permiten organizar los elementos de la interfaz y controlar la inter actuación con el usuario. Los componentes de proceso de usuario resultan especialmente útiles cuando la inter actuación del usuario sigue una serie de pasos predecibles, como al utilizar un asistente para realizar una tarea determinada. En la figura 7 se muestran los tipos de componentes presentes en la capa de presentación.

Si tomamos el caso de una aplicación comercial, son necesarias dos interfaces de usuario: una para el sitio *web* de comercio electrónico que utiliza el cliente y otra para las aplicaciones basadas en formularios de *Windows Forms*. Ambos tipos de usuario realizan tareas similares a través de estas interfaces.

2.2.3. Diseño de capas empresariales

La parte más importante de la aplicación es la funcionalidad que proporciona. Una aplicación realiza un proceso empresarial que consta de una o varias tareas. En los casos más simples, cada tarea se puede encapsular en un método de un componente *.net* y llamar de forma sincrónica o asincrónica. Para los procesos empresariales más complejos que requieren varios pasos y transacciones de ejecución larga, la aplicación necesita disponer de un modo de organizar las tareas empresariales y almacenar el estado hasta que el proceso se haya completado.

Se puede diseñar la lógica en las capas empresariales para su uso directo por parte de componentes de presentación o su encapsulación como servicio y llamada a través de una interfaz de servicios, que coordina la conversación asincrónica con los llamadores del servicio e invoca el flujo de trabajo o los componentes empresariales. La parte principal de la lógica empresarial se suele denominar lógica de dominio. Los componentes empresariales también pueden realizar solicitudes de servicios externos, en cuyo caso tal vez sea preciso implementar agentes de servicios para administrar la

conversación requerida para la tarea empresarial específica realizada por cada uno de los servicios que necesita utilizar.

2.2.4. Diseño de capa de datos

Casi todas las aplicaciones y servicios necesitan almacenar y obtener acceso a un determinado tipo de datos.

Al trabajar con datos se debe determinar:

- a) Almacén de datos que se utiliza.
- b) Diseño de componentes utilizados para obtener acceso al almacén de datos.
- c) Formato de datos pasados entre componentes y el modelo de programación necesario para ello.

La aplicación o servicio puede disponer de uno o varios orígenes de datos, los cuales pueden ser de tipos diferentes. La lógica utilizada para obtener acceso a los datos de un origen de datos se encapsula en componentes lógicos de acceso a datos que proporcionan los métodos necesarios para la consulta y actualización de datos. Los datos con los que la lógica de la aplicación debe trabajar están relacionados con entidades del mundo empresarial que forman parte de la empresa. En determinados escenarios, se puede disponer de componentes personalizados que representan estas entidades, mientras que en otros se puede decidir trabajar con datos utilizando directamente conjuntos de datos *ado.net* o documentos *XML*.

La mayoría de las aplicaciones utilizan una base de datos relacional como almacén principal de los datos de la aplicación. También se puede utilizar el almacén de *web*

exchange server, bases de datos heredadas, el sistema de archivos o servicios de administración de documentos.

Cuando la aplicación recupera datos de la base de datos, puede hacerlo utilizando un formato de conjunto de datos *DataReader*. A continuación los datos se transfieren entre las capas y los distintos niveles de la aplicación y, finalmente, uno de los componentes los utiliza. Tal vez se desee utilizar formatos de datos diferentes para recuperar, pasar y utilizar datos; por ejemplo, se puede utilizar los datos de un conjunto de datos para llenar las propiedades de un objeto de entidad personalizado. No obstante, se debería intentar mantener una coherencia en cuanto al tipo de formato utilizado, ya que mejorará probablemente el rendimiento y la facilidad de mantenimiento de la aplicación para presentar sólo un conjunto limitado de formatos, evitando así la necesidad de capas de traducción adicionales y de familiarizarse con *API* diferentes.

2.3. Diseño de aplicaciones y servicios

2.3.1. Diseño de la directiva de seguridad

La directiva de seguridad se ocupa de la autenticación, autorización, comunicación segura, auditoría y administración de perfiles, tal como muestra la figura 8.

Figura 8. Aspectos de la directiva de seguridad



Fuente: *Microsoft Corporation*, Diseño de aplicaciones y servicios. <http://msdn.microsoft.com/es-es/library/ms978340.aspx>.

2.3.1.1. Principios generales sobre seguridad

Existen ciertos principios generales sobre seguridad que se deben tener en cuenta a la hora de desarrollar una directiva de seguridad. Siguiendo las siguientes directrices:

- a) Siempre que sea posible, se debe recurrir a sistemas de seguridad que se hayan comprobado y demostrado su eficacia en lugar de generar su propia solución personalizada. Si se realiza un desarrollo personalizado de la infraestructura de seguridad, es necesario validar su enfoque y técnicas mediante auditoría con expertos y organizaciones que se dedican a la revisión de la seguridad, antes y después de su implementación.
- b) Nunca se debe confiar en las aportaciones externas. Es necesario validar todos los datos que introduzcan los usuarios o envíen otros servicios.
- c) Considerar por principio que los sistemas externos no son seguros. Si la aplicación recibe datos confidenciales sin cifrar desde un sistema externo, se debe asumir que dicha información no es segura.
- d) Aplicar el principio del menor privilegio. No habilitar más atributos en las cuentas de servicios que los que resulten estrictamente necesarios para la aplicación.
- e) Reducir el área de superficie. El riesgo se incrementa según aumenta el número de componentes y datos que se hayan expuesto a través de la aplicación y, por lo tanto, se deberá exponer únicamente la funcionalidad que se asume que otros van a utilizar.
- f) Establecer como predeterminado un modo seguro. No habilitar servicios, tecnologías y derechos de cuenta que no sean absolutamente necesarios. Cuando se

implemente la aplicación en equipos cliente o servidor, la configuración predeterminada de ésta deberá ser segura.

- g) No confiar en la seguridad a través del ocultamiento. El cifrado de los datos implica disponer de claves y de un algoritmo de cifrado demostrado. El almacenamiento de los datos seguros evitará el acceso a ésta en cualquier circunstancia. No se puede considerar seguridad la mezcla de diversas cadenas, el almacenamiento de la información en rutas de archivo inesperadas y demás técnicas similares.
- h) Seguir los principios de *STRIDE*. (*STRIDE* responde a las siglas inglesas de simulación, alteración, repudio, revelación de información, denegación de servicio y elevación de privilegios). Todas estas son clases de vulnerabilidades de la seguridad contra las que un sistema se debe proteger.
- i) Realizar la comprobación desde la misma puerta. No permitir que los procesos vayan más allá del lugar para el que los usuarios están autorizados.
- j) Bloquear el sistema interna y externamente: los usuarios y operadores internos pueden representar un riesgo igual que los intrusos externos.

2.3.1.2. Autenticación

La autenticación se define como identificación segura, que básicamente quiere decir que se dispone de un mecanismo para identificar con seguridad a los usuarios que se adecuan a los requisitos de seguridad de la aplicación. La autenticación se debe implementar en la capa de la interfaz de usuario para proporcionar funciones de autorización, auditoría y personalización.

2.3.1.3. Flujo de identidad entre los niveles

Cuanto más lejos del usuario se encuentra una parte de la funcionalidad, menos significativa se vuelve la identidad de éste. En una solución basada en servicios, algunas actividades ni siquiera las inicie un usuario. El objetivo de su diseño es reducir la relevancia del usuario cuanto más lejos de la interfaz de usuario esté la actividad.

Puede que se necesite establecer un flujo de las identidades de los llamadores originales (usuarios o servicios) a través de las capas de la aplicación para realizar la autorización o auditoría. La identidad puede ser la de un llamador original (usuario o servicio), o bien una cuenta de servicio de un nivel de aplicación. Para establecer el flujo de la identidad, se puede permitir que el mecanismo de comunicación establezca el flujo del contexto de seguridad (por ejemplo, mediante el uso de la delegación de *Kerberos* junto con la interacción remota de *DCOM*), se puede pasar símbolos (*tokens*) o vales de autenticación, o bien el *Id.* O las credenciales del usuario.

2.3.1.4. Autorización

El aspecto de la autorización de la directiva de seguridad se ocupa de la identificación de las acciones permitidas para cada principal de seguridad autenticado. En otras palabras, la directiva de seguridad determina quién puede hacer qué. Para determinar la directiva de autorización, es necesario tener en cuenta dos factores principales:

- a) Los permisos y derechos de usuario
- b) La seguridad de acceso al código

Los permisos y derechos de usuario determinan lo que se permite hacer en una cuenta de usuario en el contexto de la aplicación. Técnicamente, el término

"permisos" se refiere a las acciones permitidas en un recurso, mientras que los "derechos" hacen referencia a las tareas del sistema que se permite realizar al usuario. Los permisos y derechos de usuario se pueden asignar de forma individual para cada usuario, si bien resultan más fáciles de administrar cuando los usuarios se organizan de una manera lógica en grupos o funciones.

La mayor parte de los recursos tienen algún tipo de lista de permisos relacionada, en la que se indican los permisos asignados a los usuarios para ese determinado recurso. Los permisos son generalmente acumulativos, por lo que un usuario que tiene permiso de "lectura" en un archivo y que se encuentra en un grupo que tiene permiso de "modificación" en ese mismo archivo, tendrá un permiso de red de "modificación". Si a un usuario, o a cualquiera de los grupos de los que este usuario es miembro, se le deniega explícitamente el acceso a un recurso, no podrá tener acceso al recurso, independientemente de los permisos que se hayan asignado a cualquier usuario o grupo.

2.3.2. Diseño de la directiva de administración operativa

La directiva de administración operativa se ocupa de la ejecución constante y diaria de la aplicación y abarca aspectos como la administración de excepciones, la supervisión, la supervisión empresarial, los metadatos, la configuración y la ubicación del servicio, tal como se muestra en la figura 9

Figura 9. Aspectos de la directiva de administración operativa.



Fuente: *Microsoft Corporation*, Diseño de aplicaciones y servicios. <http://msdn.microsoft.com/es-es/library/ms978340.aspx>.

2.3.2.1. Administración de excepciones

La administración de excepciones incluye la detección y generación de excepciones, el diseño de éstas, el flujo de información de las mismas y la publicación de información de las excepciones a diversos usuarios. Todas las aplicaciones deben implementar algún tipo de control de las excepciones para detectar errores en tiempo de ejecución. Las excepciones se deben detectar y resolver si es posible. Si no se puede resolver un estado de error, la aplicación deberá mostrar un mensaje descriptivo para el usuario y proporcionar algún medio para el registro o publicación de la información de la excepción para la depuración.

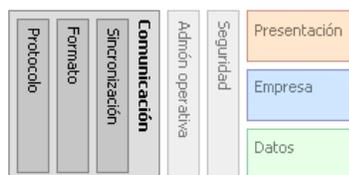
2.3.2.2. Ubicación de servicios

En las llamadas a servicios remotos, es necesario determinar dónde están situados los objetos y servicios externos de *.net* que pueden procesar la solicitud.

2.3.3. Diseño de la directiva de comunicaciones

La directiva de comunicaciones define la forma en que los componentes de la aplicación se comunicarán. Esta directiva trata cuestiones como la sincronización de la comunicación, el formato y el protocolo, tal como se muestra en la figura 10.

Figura 10. Aspectos de la directiva de comunicaciones



Fuente: *Microsoft Corporation*, Diseño de aplicaciones y servicios. <http://msdn.microsoft.com/es-es/library/ms978340.aspx>.

2.4. Implementación física y requisitos operativos

2.4.1. Implementación de los componentes de la aplicación

Es importante hacer énfasis que las capas constituyen simplemente una forma adecuada de describir los tipos de funcionalidad de la aplicación. Se trata más bien de divisiones conceptuales que de un patrón de implementación física. La forma en que las capas físicas de la aplicación se implementan en los niveles se basa en el modo de interacción de las capas entre sí y en los requisitos de los que disponen desde el punto de vista de la seguridad, las operaciones y la comunicación.

Finalmente, la aplicación se instalará en una infraestructura física. En algunos casos, el arquitecto podrá definir la infraestructura física, pero en muchos otros, el departamento de tecnologías de la información será el que la establezca. Los patrones de implementación física se suelen decidir mediante una negociación entre el departamento de tecnologías de la información y los desarrolladores de la aplicación motivados por el arquitecto de la solución.

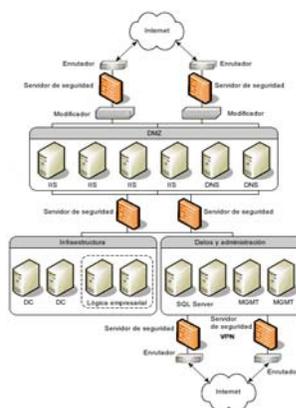
En cualquier escenario de implementación, se debe:

- a) Conocer desde un principio el entorno de implementación físico de destino, desde la fase de planeamiento del ciclo de vida.
- b) Establecer claramente qué restricciones del entorno condicionan el diseño del *software* y la toma de decisiones relativas a la arquitectura.
- c) Transmitir con claridad qué decisiones acerca del diseño del *software* requieren determinados atributos de infraestructura.

2.4.2. Entornos físicos de implementación

Dichos entornos varían dependiendo de varios factores: tipo de aplicación que se implemente, base de usuario de la aplicación, escalabilidad, requisitos de rendimiento, directivas de organización, etc.

Figura 11. **Arquitectura de *Internet datacenter***



Fuente: *Microsoft Corporation*, Diseño de aplicaciones y servicios. <http://msdn.microsoft.com/es-es/library/ms978340.aspx>.

Al igual que una aplicación consta de componentes y servicios, la infraestructura que la aloja se puede considerar como una serie de unidades de creación de infraestructura, denominadas niveles físicos. Estos niveles representan las divisiones físicas que existen entre los componentes de la aplicación y pueden o no asignarse directamente a los niveles lógicos utilizados para abstraer los distintos tipos de funcionalidad de la aplicación. Los niveles físicos pueden estar separados por servidores de seguridad u otras medidas de seguridad para crear diferentes unidades de confianza o contextos de seguridad. Existen dos familias principales de niveles físicos: baterías y *clusters*.

Las baterías están compuestas por conjuntos de servidores ampliables y configurados de idéntico modo que comparten la carga de trabajo. Los *clusters* son conjuntos de equipos especializados que controlan un recurso compartido.

2.4.3. Requisitos Operativos

Consiste en proporcionar las técnicas de diseño y las prácticas que permitirán obtener los requisitos operativos (no funcionales) para la aplicación y los servicios. Entre estos requisitos se incluyen los niveles de escalabilidad, disponibilidad, mantenimiento, seguridad y facilidad de uso que debe obtener la aplicación. Estos factores pueden afectar al diseño de las directivas de la aplicación, aunque también pueden influir en el modo de diseño de la lógica de la aplicación.

En algunos casos, el cumplimiento con algunos requisitos supondrá la aparición de retos para llevar a cabo otros. Por ejemplo, es frecuente reducir la facilidad de uso de una aplicación para mejorar la seguridad. Es importante otorgar prioridad a las características de la aplicación que admiten los requisitos operativos desde un primer momento del ciclo de vida, por lo que estos equilibrios y decisiones se pueden tener en cuenta en la implementación de la aplicación desde un primer momento.

3. FASE DE ENSEÑANZA APRENDIZAJE

3.1. Justificaciones del sitio *web*

3.1.1. Justificación general de la implementación

En la actualidad, es difícil que una universidad pueda abarcar totalmente el número de áreas de conocimiento existentes y debido a esto, desarrollan sitios *web* los cuales sirven al estudiante así como al docente para informarse y para la transferencia de información, siendo este un sitio actualizado en cualquier momento.

El sitio *web* de la unidad de EPS tiene como finalidad establecer las normas y procedimientos para hacer posible brindar conocimientos e información académica, a través de los cuales permitan perfeccionar y elevar el conocimiento de los estudiantes, Por esta razón se optó por la realización del sitio para desarrollar la solución de la propuesta planteada y permitir que dicha *web* sea realizada por el procedimiento de EPS ya que el departamento no cuenta con presupuesto para el desarrollo de este proyecto.

3.1.2. Justificación técnica

- a) Sistema de información robusto y actualizable, capaz de llevar el control y administración de la información digital.
- b) Ser una herramienta de comunicación rápida, confiable y actualizada, aparte de administrable por personal del propio departamento de EPS.

- c) Que los procedimientos se conviertan en una herramienta para el desarrollo de un mejor nivel para los estudiantes que finalizan la carrera, y para estudiantes de práctica.

3.1.3. Justificación económica

- a) El proyecto ha sido realizado por medio de un trabajo de EPS, sin incurrir en gastos de una empresa desarrolladora.
- b) El sitio *web* de la unidad de EPS se ha colocado dentro de la red local de la facultad de ingeniería para evitar caer en gasto al departamento.
- c) Se ha utilizado una máquina donada por la facultad de ingeniería la cual ha sido actualizada para el buen funcionamiento de la página.
- d) Se dará soporte al sitio hasta lograr la familiaridad del administrador nombrado por el departamento de EPS.

3.1.4. Justificación social

- a) El desarrollo del programa permite que los estudiantes tengan mayores oportunidades de conocer los temas de actualidad y hacia dónde se dirige la tecnología, en base a trabajos realizados por estudiantes que terminan su carrera.
- b) Llegar a la mayoría de personas interesadas y servir de información para cada estudiante que se encuentra en sus últimos trámites para optar por el título.
- c) La población estudiantil recibe información de actividades donde pueda participar y optar por sus trabajos de graduación.

- d) Mantener comunicación constante entre estudiantes y catedráticos del área de EPS.

3.2. Tecnología utilizada

3.2.1. Tecnología de desarrollo

La tecnología que se ha utilizado para este proyecto es *visual studio.net*, para lo que es el diseño y desarrollo de la aplicación. En la actualidad existen un sin fin de sitios en los cuales se puede encontrar código libre.

Dentro de las tecnologías y *software open source* que se han utilizado se tiene el *ajax* en el modelo actual, la *web* es un lugar muy hostil para desarrollar aplicaciones. Cada vez que el usuario ejecuta una acción (un clic, la presión de una tecla, el arrastre de un objeto) el navegador debe solicitar datos a otra computadora (el servidor) a través de *Internet*, para luego regenerar la página que el usuario está viendo. De esta forma, la interacción se transforma en un pimpón de datos que hacen imposible cualquier actividad continua, como un juego o una aplicación.

Para sortear este inconveniente, varias empresas desarrollaron plataformas que se cargan dentro del navegador, como *Macromedia Flash* o *Java*, previa descarga de los *plugin* respectivos y previo pago de las licencias para el desarrollo. Pero, ¿qué pasaría si tuviéramos estas herramientas ya disponibles en los computadores de cada usuario y en forma nativa? El modelo de *ajax* propone cargar y renderizar una página, luego sirviéndose de las tecnologías antes mencionadas y presentes en la gran mayoría de los navegadores. Mantenerse en esa página, mientras *scripts* y rutinas van al servidor buscando, en *background*, los datos que son usados para actualizar la página solo renderizando la página y mostrando u ocultando porciones de la misma.

Los pasos para configurar esta tecnología son los siguientes:

a) Agregar una referencia al archivo *ajax.dll*, este archivo se puede descargar desde <http://ajax.schwarz-interactive.de/download/ajax.zip>;

b) Configurar el *HttpHandler* en el archivo *web.config*, esta configuración se describe en el código descrito a continuación.

```
<system.web>
<httpHandlers>
<add verb="POST,GET" path="ajaxwrapper/*.ashx"
type="Ajax.PageHandlerFactory, Ajax" />
</httpHandlers>
</system.web>;
```

c) En el *pageload* del *form* se agrega el llamado a esta función *Ajax.Utility.RegisterTypeForAjax(this.GetType());*

d) Para hacer llamados asincrónicos al servidor se debe marcar las funciones con el siguiente atributo

```
[Ajax.JavascriptMethod()]
public int ServerSideAdd(int firstNumber, int secondNumber)
{ return firstNumber + secondNumber;}
```

La envoltura creará automáticamente el código *java script* con los parámetros y será llamada desde el cliente sin hacer *postback*;

e) Se puede llamar estas funciones desde el *java script* en el cliente.

```
alert(ServerSideAdd(100,99)); o
ServerSideAdd(100,99, ServerSideAdd_CallBack);
function ServerSideAdd_CallBack(result){
alert(result);}.
```

Para la base de datos se ha utilizado *Microsoft SQL 2000* y queda preparada para una futura migración a *SQL Server 2007*.

3.2.2. Tecnología de edición de texto

3.2.2.1. Editor de texto *HTML WYSIWYG* de *TINYMCE*

TinyMCE es un editor *HTML* capaz de convertir un *textarea* de un formulario en campos *WYSIWYG* para poder incluir etiquetas *HTML* dentro de los campos de texto.

Características

- a) Fácil de integrar en páginas *web*, ya que solo tiene dos líneas de código.
- b) Se puede personalizar a través de temas y *plugins*.
- c) También se pueden instalar paquetes de idiomas.
- d) Es compatible con la mayoría de los navegadores como *firefox*, *Internet explorer*, *opera* y *safari*.
- e) Con el compresor *GZip* para *PHP/.NET/JSP/Coldfusion*, hace que *TinyMCE* sea un 75% más pequeño y mucho más rápido de cargar.
- f) Se puede utilizar *ajax* para guardar y cargar el contenido.

Integración de *TinyMCE*

Para poder utilizar *TinyMCE* en las páginas *web*, el navegador tiene que ser compatible y tener *java script* habilitado. Luego se debe bajar y guardar los archivos en el servidor de la *web* y entonces ya será posible poder utilizar en la *textarea* de los formularios. En la página *web* de EPS, primero se incluye la librería *tiny_mce.js* incluyendo el archivo externo de código *java script*.

```
<script language="javascript" type="text/javascript"
src="/tinymce/jscripts/tiny_mce/tiny_mce.js"></script>
```

A continuación se debe de inicializar *TinyMCE* para convertir los *textarea* en campos de texto *WYSIWYG* editables.

```
<script language="javascript" type="text/javascript">
tinyMCE.init({ mode : "textareas",
  theme : "simple"
});</script>
```

Ejemplo de integración de *TinyMCE*

```
<html>
<head>
  <title>Ejemplo TinyMCE</title>
  <script language="javascript" type="text/javascript"
src="/tinymce/jscripts/tiny_mce/tiny_mce.js"> </script>
  <script language="javascript" type="text/javascript">
    tinyMCE.init({ mode : "textareas",
      theme : "advanced"
    }); </script>
</head>
<body>
```

```
<form method="post" name="tinymce">
  <textarea name="texto" cols="50" rows="15"></textarea>
</form>
</body>
</html>
```

En este trozo de código primero se ha incluido la librería *tiny_mce.js* dentro de las etiquetas *<head>*. También dentro de estas etiquetas se ha inicializado *TinyMCE* para que en el *textarea* del formulario se convierta en un campo de texto *WYSIWYG*.

3.3. Recursos disponibles

3.3.1. Recurso humano

- a) Personal del departamento de EPS.
- b) Personal de estudios profesionales supervisados (EPS).
- c) Apoyo del director de la escuela de Sistemas, Ing. Marlon Perez Turk.
- d) Apoyo de directora de EPS Inga. Norma Sarmiento.
- e) Asesor supervisor por parte de la escuela de Ciencias y Sistemas Ing. Jorge Armin Mazariegos Rabanales.
- f) Asesor supervisor de la unidad de EPS. Ing. Floriza Ávila.

3.3.2. Recursos de materiales

- a) Equipo de cómputo con acceso a *Internet*
- b) Impresora
- c) Equipo de oficina

3.3.3. Recurso financiero

- a) El equipo donde se llevará a cabo el diseño y desarrollo del proyecto será proporcionado por el estudiante EPSISTA a cargo del proyecto.
- b) No es necesario financiar el dominio del sitio *web* ya que para esto se utilizarán los recursos de la facultad de ingeniería.
- c) No es necesario financiar el hospedaje del sitio *web* ya que se utilizan los recursos de la facultad de ingeniería.
- d) El financiamiento para la licencia de *Microsoft SQL* Ha sido donado por la Escuela de Ciencias y Sistemas de la facultad de Ingeniería.
- e) El financiamiento para *Visual Studio .net* ha sido donado por la Escuela de Ciencias y Sistemas.

3.3.4. Presupuesto

Costos y gastos en que se incurrieron en el proyecto se detallan a continuación:

Tabla I. Presupuesto

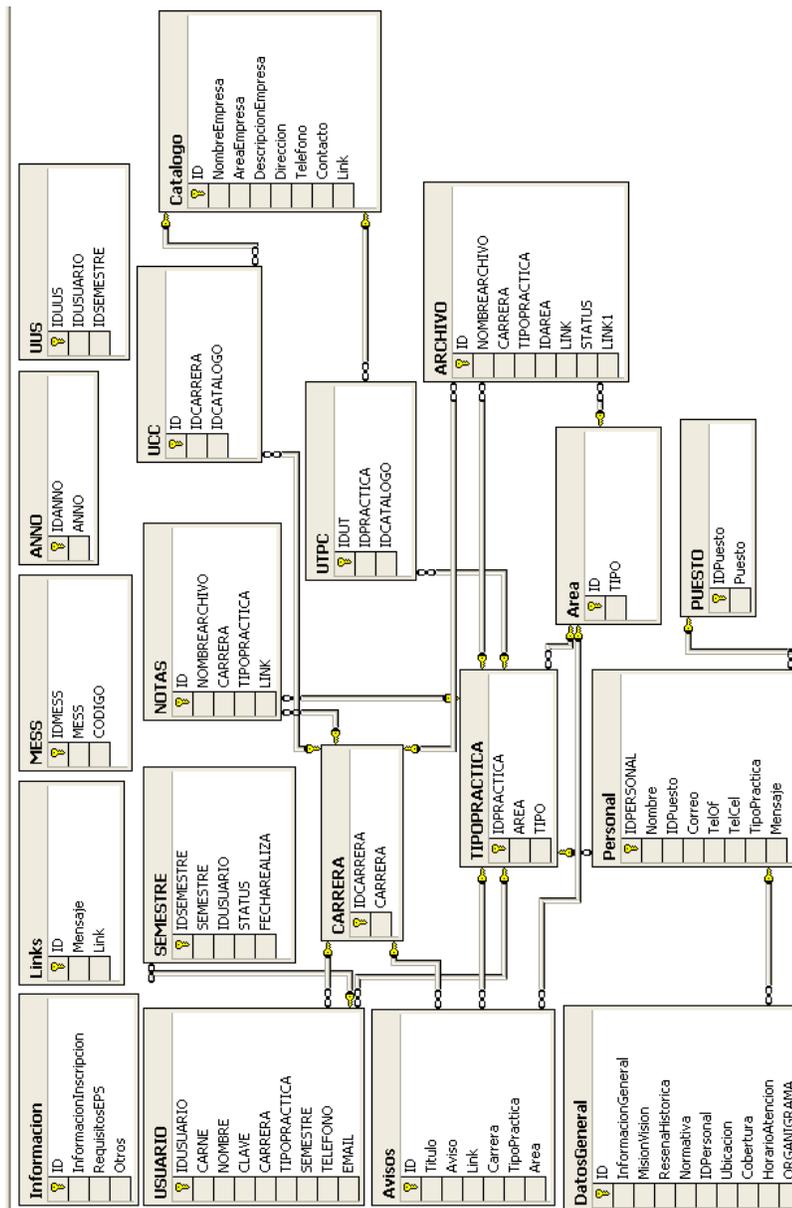
Cantidad	Descripción	Costo	Sub-Total	Notas
1	Equipo de computo	Q9,800.00	Q9,800.00	
1	<i>UPS</i>	Q300.00	Q300.00	
12	Servicio de <i>Internet</i>	Q399.35	Q4,792.20	Mensual
1	Licencia de <i>Microsoft Office</i>	Q2,000.00	Q2,000.00	
1	Licencia de <i>Visual Studio .Net</i>	Q4,425.25	Q4,425.25	
1	Licencia de <i>Microsoft SQL Server</i>	Q3,000.00	Q3,000.00	
1	Licencia de <i>Microsoft Visio</i>	Q1,015.25	Q1,015.25	
12	Sueldos de jefe de proyecto	Q8,000.00	Q96,000.00	Mensual
12	Sueldos de analista programador	Q7,000.00	Q84,000.00	Mensual
1	Sueldo de encargado de pruebas y calidad de <i>software</i>	Q5,000.00	Q5,000.00	
1	Sueldo de un documentador	Q4,000.00	Q4,000.00	
1	Sueldo de diseñador gráfico	Q5,000.00	Q5,000.00	
1	Dominio para la pagina <i>web</i>	Q 70.3	Q 70.30	Anual
12	Servicio de hospedaje de la página.	Q 31.35	Q 375.90	Mensual
Total			Q219,778.90	

Fuente: elaboración propia.

3.4. Diagramas del proyecto

3.4.1. Diagrama entidad relación

Figura 12. Diagrama entidad relación

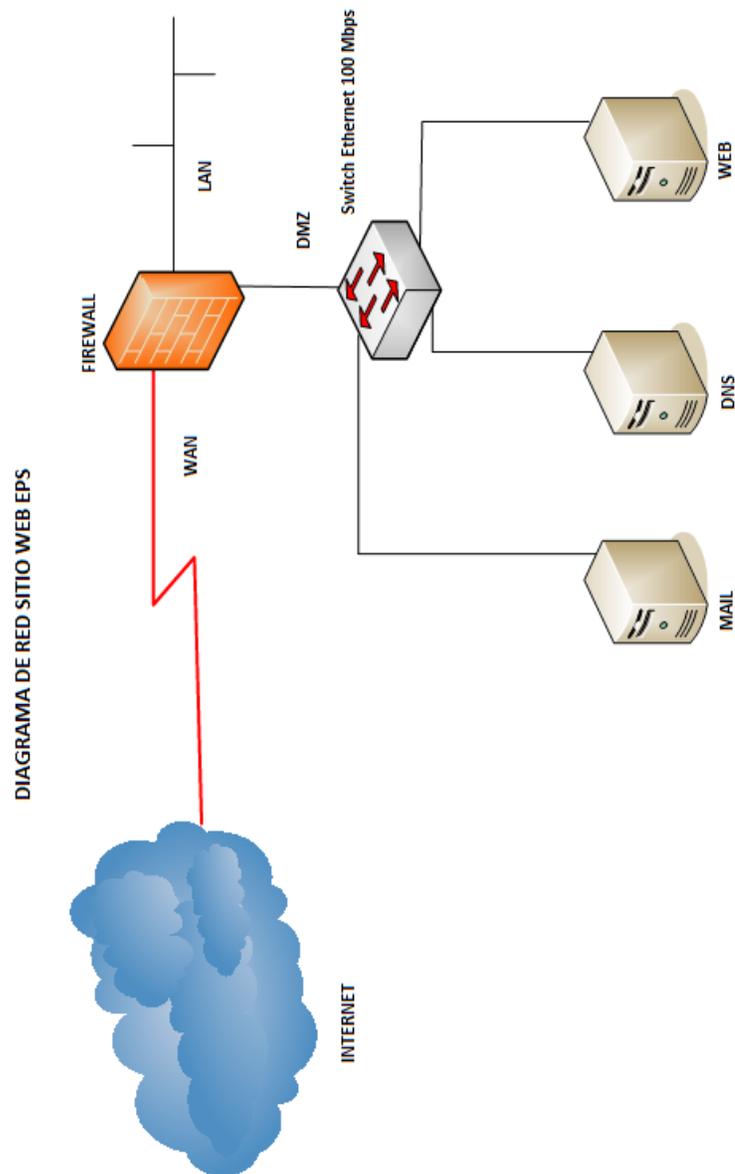


Fuente: elaboración propia.

3.4.2. Diagramas de red

3.4.2.1. Diagrama de red1

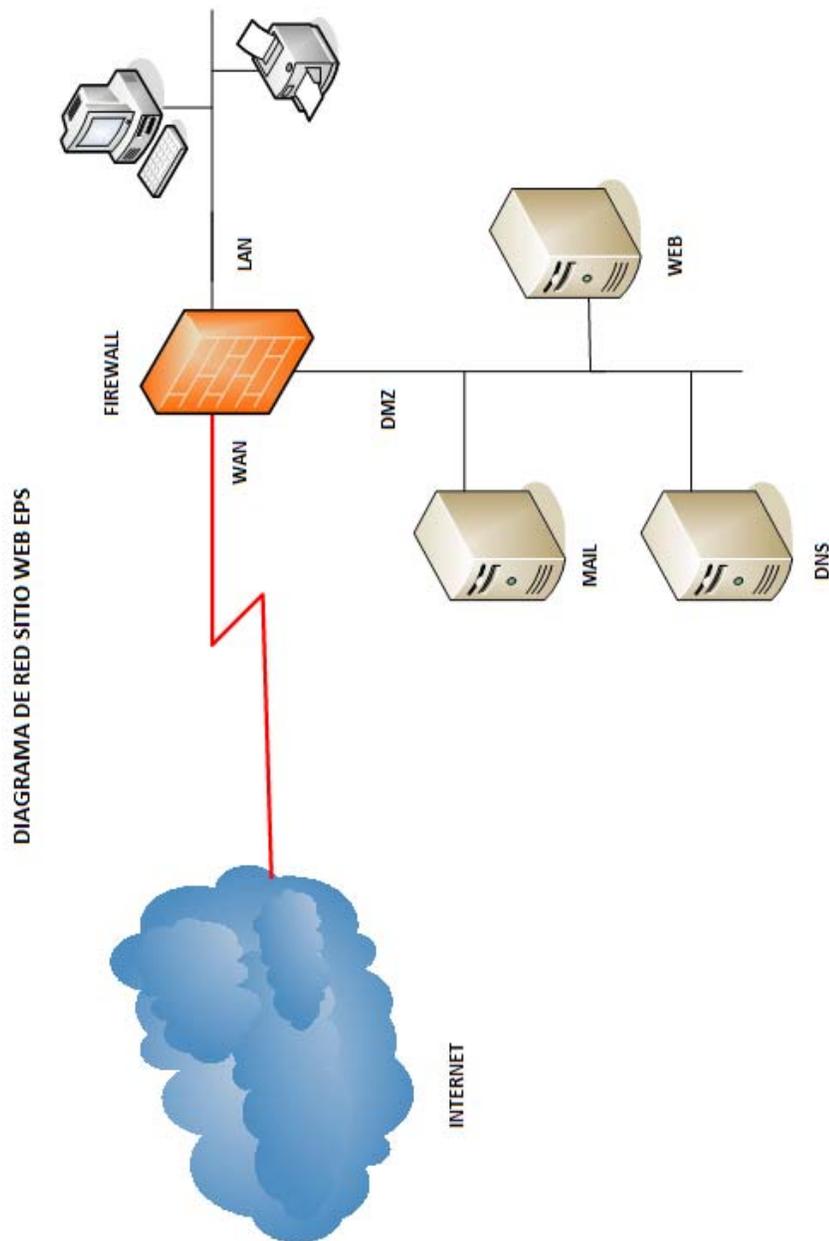
Figura 13. Diagrama de red Número 1



Fuente: elaboración propia.

3.4.2.2. Diagrama de red 2

Figura 14. Diagrama de red número 2



Fuente: elaboración propia.

c) Seleccionar opción “Agregar o quitar componentes de *Windows*”;

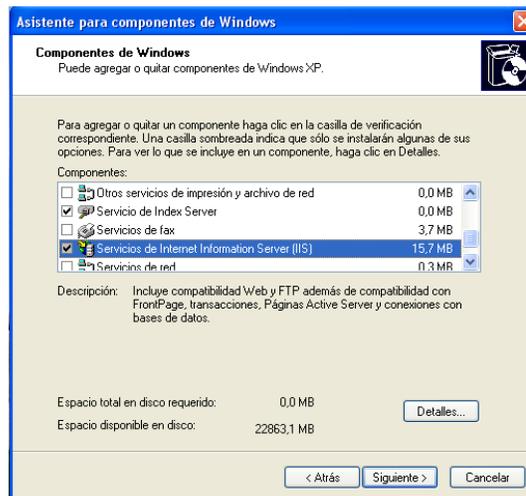
Figura 16. **Agregar o quitar componentes de *Windows***



Fuente: elaboración propia.

d) Al mostrar la lista de servicios seleccionar el servicio deseado en este caso “Servicio de *Internet Information Server (IIS)*”, luego presionar siguiente;

Figura 17. **Asistente para componentes de *Windows***



Fuente: elaboración propia.

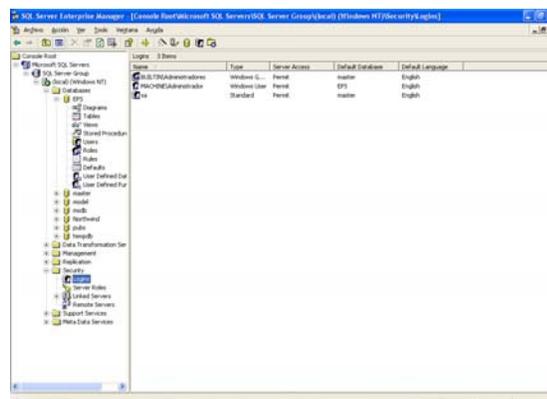
- e) Presionar el botón finalizar y el servicio quedará instalado correctamente. Este servicio permite que el equipo sea visto por una red local o desde *Internet* valiéndose de la dirección de salida designada por un servidor de *DNS* mediante el uso de un navegador de páginas *web*. El *IIS* crea en el disco raíz un directorio llamado *INETPUB* el cual contendrá documentos de administración y directorios donde reside el sitio *web* o los directorios virtuales. El sitio *web* deberá ser almacenado en la carpeta *WWWROOT* la cual es tomada como la carpeta principal o raíz del sitio. En *Windows 2003 SERVER* es necesario habilitar en agregar y quitar programas en componentes de *Windows* la opción de *ASPX* y que no lo trae habilitado y por consiguiente no ejecuta código *ASPX*.

4.1.1.2. Guía para la configuración de servidor en *trusted source*

Configuración *trusted source* en *SQL*:

- a) Abrir el *SQL* server *enterprise*;
- b) Navegar hacia el nodo de seguridad bajo el nodo (local)(*Windows NT*);

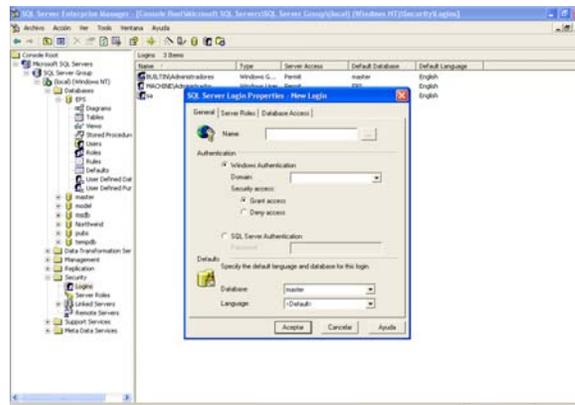
Figura 18. *SQL enterprise manager*



Fuente: elaboración propia.

- c) Clic derecho sobre el Nodo *log* del Nodo de seguridad y en nuevo logueo;

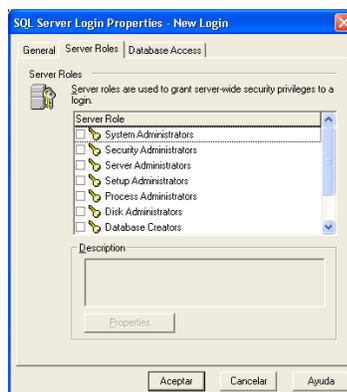
Figura 19. **Propiedades de logueo de SQL**



Fuente: elaboración propia.

- d) Buscar nombre de usuario administrador y seleccionar base de datos por *default*;
- e) seleccionar la pestaña *database roles*;

Figura 20. **Roles de base de datos**

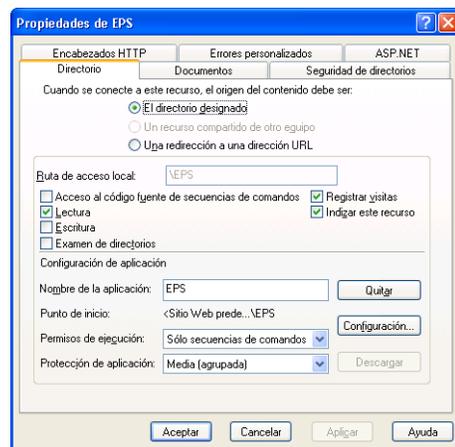


Fuente: elaboración propia.

- f) seleccionar los roles para el administrador;

- b) Navegar por la ventana *web* predeterminada hasta llegar al nodo del proyecto EPS, dar clic derecho y seleccionar propiedades;

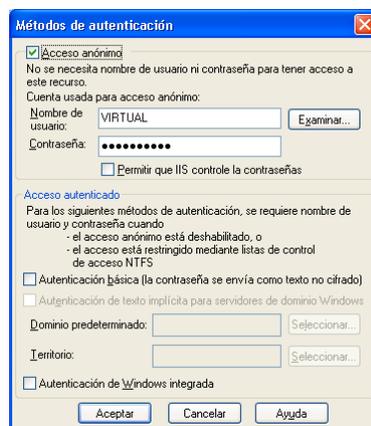
Figura 23. **Propiedades de proyecto EPS**



Fuente: elaboración propia.

- c) Clic en la pestaña seguridad de directorios, dar clic en el botón modificar del control de autenticación y acceso anónimo;

Figura 24. **Métodos de autenticación**



Fuente: elaboración propia.

- d) En método de autenticación seleccionar opción acceso anónimo y en nombre de usuario seleccionar usuario con permisos de acceder a pagina *web*, seleccionar opción permitir que *IIS* controle contraseñas e ingresar la contraseña de usuario.

Configuración *asp.net* con *SQL* y *IIS* para acceso *trusted source*:

- a) Abrir el archivo *machine.config* el cual se encuentra en la siguiente ubicación *C:\WINNT\Microsoft.NET\Framework\v1.0.3705\CONFIG*;
- b) Buscar el tag *<identity>*, y cambiar el atributo *impersonate* con el valor *true*;
- c) Verificar que los atributos de usuario y *password* estén vacios, guardar el archivo.

4.1.2. Guía administrativa

4.1.2.1. Para ingresar a la *web* de administración

Para ingresar a la *web* de administración de la página se debe de navegar por la pestaña área administrativa y luego dar clic en la opción administración *web*.

Figura 25. Área administrativa EPS



Fuente: elaboración propia.

Colocar credenciales proporcionadas al director de EPS.

Figura 26. **Página de logueo**



Fuente: elaboración propia.

4.1.2.2. **Modificar datos de director**

Se da clic en la pestaña de área administrativa, dar clic en la opción de “Modificar Datos Director”. Luego se ingresa en la ventana de modificación de director.

Figura 27. **Modificar datos director**



Fuente: elaboración propia.

Se puede observar los datos del director que se pueden actualizar, luego de realizar los cambios, se tienen 2 opciones, la opción guardar y la opción reiniciar, en guardar si se presiona en este botón se realizan los cambios que se hayan realizado, en la opción reiniciar se dejan los datos como originalmente se encuentran en la base de datos.

4.1.2.3. Modificar datos de personal docente

Situarse en pestaña área administrativa y elegir “Modificar Personal Docente”.

Figura 28. Modificación de datos personal



EDITAR	Borrar	NOMBRE DOCENTE	CORREO DOCENTE	TELEFONO OFICINA	TELEFONO CELULAR
		Coordinador AINd	coordinador@industria.com	54	45
		Coordinador AInf	coordinador@infraestructura.com	45	45
		Coordinador ATecnologia	coordinador@tecnologia.com	54	54
		Coordinador Practica Inicial	coordinador@practicasini	41	41
		Coordinador Practica Inter	galicia.walter@telgua.com.gt	42	54
		Coordinador Practica Final	lopez.carlos@telgua.com.gt	43	55
		Secretaria	secretaria@secretaria	44	56
		Asesor	asesor@eps	47	47
		Catedratico	catedratico@pi.com	85	85

Fuente: elaboración propia.

Ya dentro de la opción de personal docente se pueden realizar 3 tipos de mantenimiento,

- a) Editar personal docente, para esta opción se da clic sobre la hoja de papel con lápiz

Figura 29. Editar personal docente



Nombre Docente:

Correo Electronico:

Tel. Oficina:

Tel. Celular:

Fuente: elaboración propia.

En esta ventana se pueden realizar los cambios necesarios de personal docente y luego dar clic en cualquiera de las 2 opciones que se tiene “Modificar” o “Cancelar” en modificar se realizarán los cambios que se han modificado y en cancelar se quedarán los datos originales de la base de datos;

- b) Borrar personal docente, para la opción de borrar se da un clic sobre la X roja de la columna borrar;
- c) Agregar personal docente, para agregar un nuevo personal se da clic sobre el botón agregar de la ventana de modificación de personal docente. Tomar en cuenta que es obligatorio ingresar los datos marcados con asterisco para evitar que el sistema de mensajes de error.

Figura 30. **Agregar nuevo personal**

The screenshot displays a web application interface for 'SISTEMA DE ADMINISTRACIÓN DE SITIO WEB DE UNIDAD EPS - SASWUEPS'. The main menu includes 'Área Administrativa', 'Área de EPS', 'Modificar Horarios de Atención', 'Subir Horario Clases', 'Ingresar', and 'Salir'. The current page is 'Inicio > Mantenimiento de Personal' and is titled 'Modificación de Personal'. The central form is 'Ingreso de Personal de EPS' and contains the following fields: 'Nombre Docente' (text input with a red asterisk), 'Correo Electronico' (text input), 'Tel. Oficina' (text input with '0' in the field), 'Tel. Celular' (text input with '0' in the field), 'Seleccione Puesto' (dropdown menu with 'Elija Un Puesto de Docente' selected), and 'Selección Área' (dropdown menu with 'Elija Un Área Para el Docente' selected). At the bottom of the form are two buttons: 'Agregar' and 'Cancelar'.

Fuente: elaboración propia.

4.1.2.4. **Modificar directores antiguos**

Se sitúa el puntero en la pestaña área administrativa y luego dar clic en “Modificar Directores Antiguos”.

Figura 31. **Modificar directores antiguos**

AGREGAR	
Editar	Borrar
NOMBRE DIRECTOR	INICIO / FIN
NORMA ILEANA SARMIENTO ZECENA DE SARMIENTO	2007 / 0
Ing. Angel Roberto Sic Garcia	2003 / 2007
ÁNGEL SIC GARCÍA	2003 / 2007
MANUEL ALFREDO ARRIVILLAGA	2000 / 2002
JUAN MERCK	1996 / 2000
PEDRO QUIROA MÉNDEZ	1992 / 1996
MANUEL ALFREDO ARRIVILLAGA	1988 / 1992
OSCAR ARGÜETA HERNÁNDEZ	1984 / 1988
ISRAEL MONTUFAR	1981 / 1984
CARLOS RAMÍREZ	1980 / 1980
BEATRIZ CHARNAUD CRUZ	1975 / 1979

Actualización: 01 de Julio del 2007.

Fuente: elaboración propia.

Ya dentro de la opción de directores antiguos se pueden realizar 3 tipos de mantenimiento,

- Editar directores antiguos: para esta opción se da clic sobre la hoja de papel con lápiz, con este procedimiento se redireccionará la pagina a la opción de modificación de directores como se muestra a continuación

Figura 32. **Edición de datos de directores**

Actualización: 01 de Julio del 2007.

Ingreso Nombre Director: NORMA ILEANA SARMIENTO ZECENA DE SARME

Ingreso Fecha de Inicio: 2007

Ingreso Fecha Final: 0

ACEPTAR CANCELAR

Fuente: elaboración propia.

En esta opción se pueden realizar los cambios necesarios de directores y luego dar clic en cualquiera de las 2 opciones que se tiene “Modificar” o “Cancelar” en modificar se realizarán los cambios que se han modificado y en cancelar se quedarán los datos originales de la base de datos.

- b) Borrar directores antiguos: para la opción de borrar se da un clic sobre la X roja de la columna borrar.
- c) Agregar director: para agregar un nuevo dato de director se da clic sobre el botón agregar de la ventana de modificación de directores antiguos

Figura 33. **Agregar datos de nuevo director**

The screenshot displays a web application interface for managing director data. At the top, there is a navigation menu with the following items: 'Area Administrativa', 'Acerca de EPS', 'Modificar Horarios de Atención', 'Subir Horario Clases', 'Ingresos', and 'Salir'. Below the navigation, the page title is 'PAGINA DE ADMINISTRACIÓN WEB' and the main heading is 'Pagina de Modificación de Directores Antiguos'. A sub-heading indicates the last update: 'Actualización: 01 de Julio del 2007'. The form contains three input fields: 'Ingreso Nombre Director', 'Ingreso Fecha de Inicio', and 'Ingreso Fecha Final'. At the bottom of the form, there are two buttons: 'AGREGAR' and 'CANCELAR'.

Fuente: elaboración propia.

Se debe tomar en cuenta que es obligatorio ingresar los datos marcados con asterisco para evitar que el sistema de mensajes de error.

4.1.2.5. **Modificar información de organigrama**

Se sitúa el puntero en área administrativa, luego dar clic en “Modificar Organigrama de EPS”.

Figura 34. **Modificar información de organigrama**



Fuente: elaboración propia.

Como se puede observar la modificación de la información del organigrama se realiza dentro de un editor *Online* el cual se asemeja a *Word*. Luego de realizar los cambios necesarios se tienen dos opciones que son “Guardar” que guarda los cambios realizados y “Reiniciar” el cual deja los valores por *default* y no realiza ningún cambio en la BDD.

Figura 35. **Modificando organigrama de EPS**

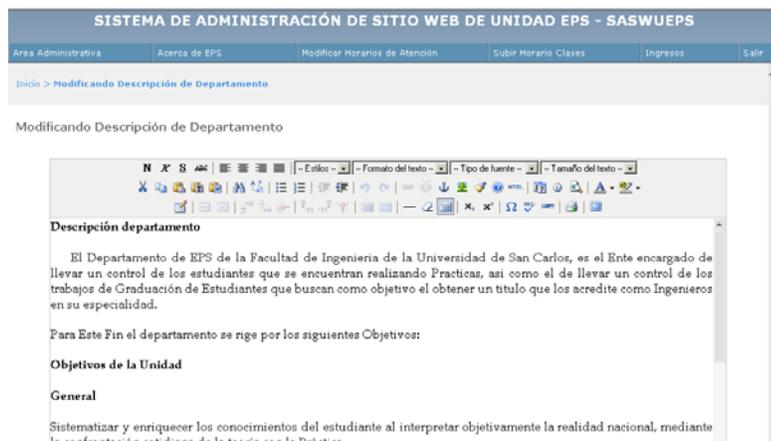


Fuente: elaboración propia.

4.1.2.6. Modificar descripción departamento de EPS

Se navega por la pestaña “Acerca de EPS” y luego dar un clic en la opción “Modificar Descripción EPSUP”. Luego se muestra la ventana de modificar descripción EPSUP en dicha ventana se muestra un editor de texto en el cual se realizan los cambios.

Figura 36. Modificar descripción de departamento



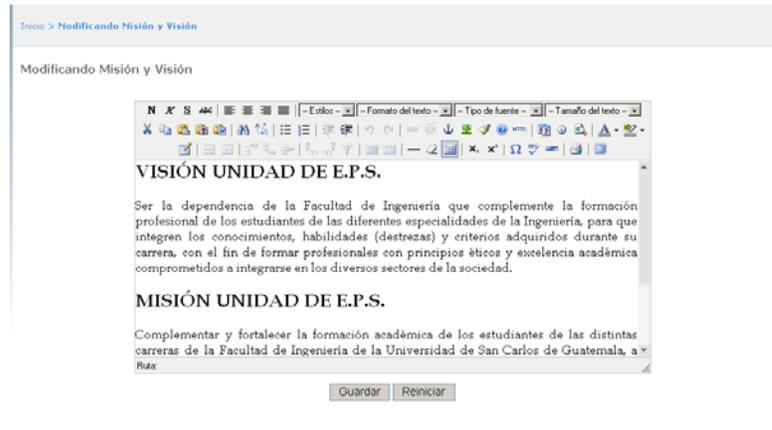
Fuente: elaboración propia.

Luego de realizar los cambios necesarios solo queda presionar en cualquiera de las opciones de los botones, puede ser “Guardar” guarda los cambios en la descripción, y “Reiniciar” que deja los valores por *default* en la base de datos

4.1.2.7. Modificar misión y visión EPS

Se navega por la pestaña “Acerca de EPS” y luego dar un clic en la opción “Modificar Misión y Visión”. Se muestra la ventana de modificar misión visión en la cual se muestra un editor de texto para la realización de cambios.

Figura 37. **Modificar información de misión y visión**



Fuente: elaboración propia.

Luego de realizar los cambios necesarios solo queda presionar en cualquiera de las opciones de los botones, puede ser “Guardar” guarda los cambios en la descripción, y “Reiniciar” que deja los valores por *default* en la base de datos.

4.1.2.8. **Modificar reseña histórica**

Se navega por la pestaña “Acerca de EPS” y luego dar un clic en la opción “Modificar Reseña Histórica”.

Se muestra la ventana de modificar reseña histórica en la cual se muestra un editor de texto en el cual se realizan los cambios.

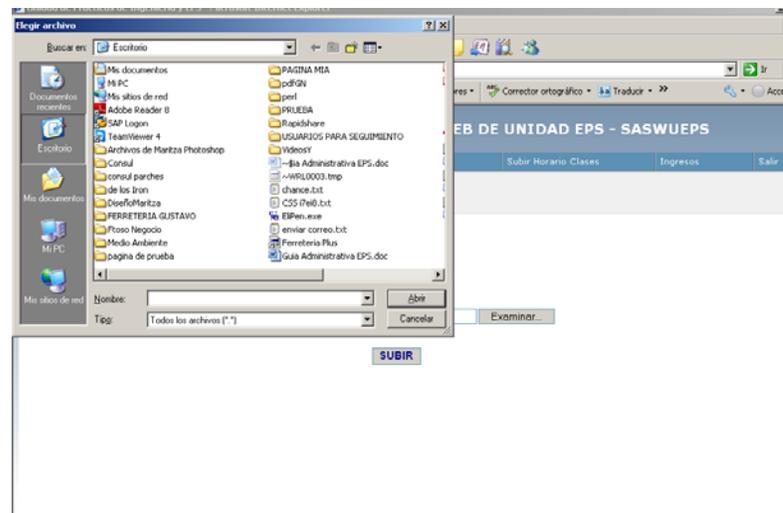
Figura 39. Subir normativo de EPS



Fuente: elaboración propia.

Luego de dar clic en examinar se mostrara una ventana en la cual se puede buscar el archivo de “Normativo de EPS” para poder subirlo

Figura 40. Seleccionando y subiendo normativo de EPS



Fuente: elaboración propia.

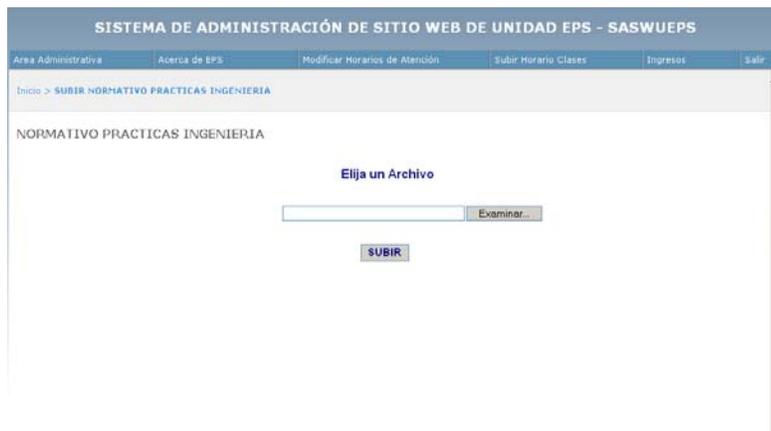
Se debe tener en cuenta que para subir normativos de EPS es necesario subirlos en formato *PDF*. Aunque la aplicación si reconoce el formato *Word* pero para seguridad del departamento es preferible manejar *PDF*. Luego de seleccionar el archivo se da clic en subir y con esto ya se ha concluido de subir el archivo.

4.1.2.10. Subir normativo de prácticas

Se navega por la pestaña “Acerca de EPS” y luego dar un clic en la opción “Subir Normativo de Prácticas”.

Luego se mostrara la ventana de subir normativo de prácticas en la cual se muestra una casilla en la cual se puede poner el direccionamiento del archivo a subir ó se puede dar clic en examinar para navegar por el equipo.

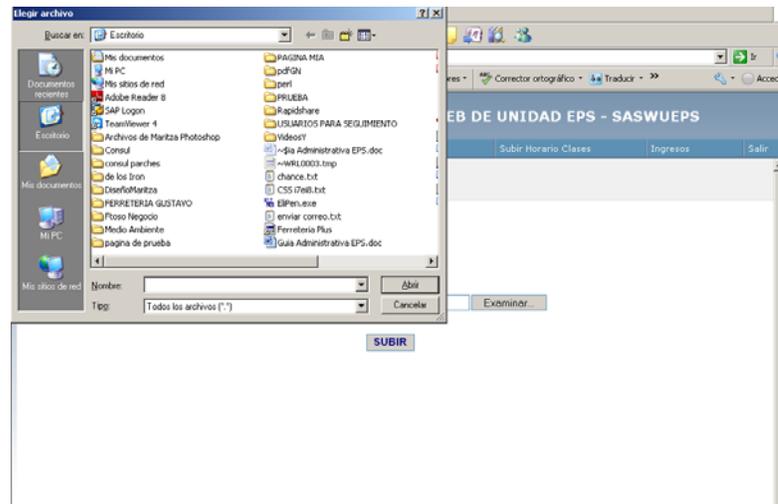
Figura 41. Subir normativo de prácticas de ingeniería



Fuente: elaboración propia.

Luego de dar clic en examinar se mostrara una ventana en la cual se puede buscar el archivo de “Normativo de Prácticas”.

Figura 42. **Seleccionando y subiendo normativo de prácticas**



Fuente: elaboración propia.

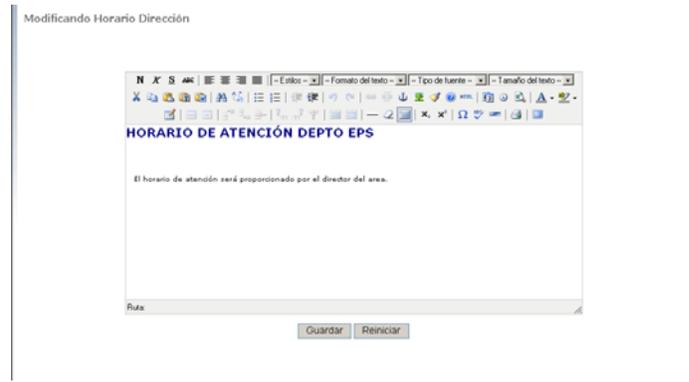
Se debe tener en cuenta que para subir normativos de prácticas actualmente se sube en formato de *Word*. Luego de seleccionar el archivo se da clic en subir y con esto ya se ha concluido de subir el archivo.

4.1.2.11. Modificar horario atención de departamento EPS

Se navega por la pestaña “Modificar Horarios de Atención” y luego dar un clic en la opción “Horario Atención Dirección”.

Luego se mostrara la ventana de modificar horario atención dirección en la cual se muestra un editor de texto en el cual se realizan los cambios.

Figura 43. **Modificar horario de atención unidad de EPS**

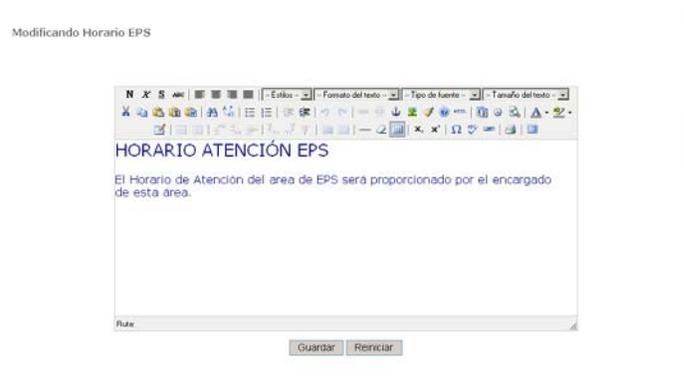


Fuente: elaboración propia.

4.1.2.12. **Modificar horario atención EPS**

Se navegar por la pestaña “Modificar Horarios de Atención” y luego dar un clic en la opción “Horario Atención EPS”. Luego se mostrara la ventana de modificar horario atención EPS en la cual se muestra un editor de texto para realizar los cambios.

Figura 44. **Modificar horario de atención de EPS**



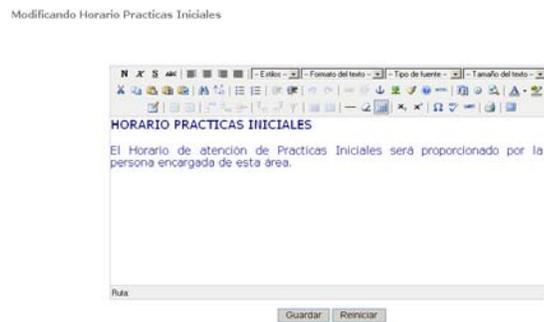
Fuente: elaboración propia.

Luego de realizar los cambios necesarios solo queda presionar en cualquiera de las opciones de los botones, puede ser “Guardar” guarda los cambios en la descripción, y “Reiniciar” que deja los valores por *default* en la base de datos.

4.1.2.13. Modificar horario atención prácticas iniciales

Se navega por la pestaña “Modificar Horarios de Atención” y luego dar un clic en la opción “Horario Atención Prácticas Iniciales”. Se muestra la ventana modificar horario atención prácticas iniciales en la cual se muestra un editor de texto en el cual se realizan los cambios.

Figura 45. Modificar horario de atención de prácticas iniciales



Fuente: elaboración propia.

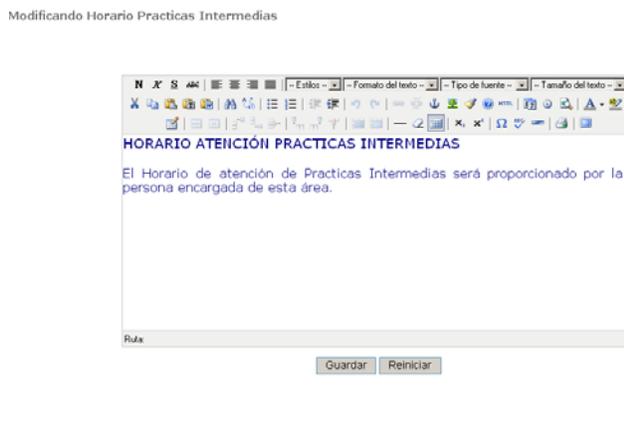
Luego de realizar los cambios necesarios solo queda presionar en cualquiera de las opciones de los botones, puede ser “Guardar” guarda los cambios en la descripción, y “Reiniciar” que deja los valores por *default* en la base de datos.

4.1.2.14. Modificar horario atención prácticas intermedias

Se navega por la pestaña “Modificar Horarios de Atención” y luego dar un clic en la opción “Horario Atención Prácticas Intermedias”. Luego se mostrara la ventana

modificar horario atención prácticas intermedias en la cual se muestra un editor de texto en el cual se puede hacer los cambios necesarios en la información.

Figura 46. **Modificar horario de atención de prácticas intermedias**



Fuente: elaboración propia.

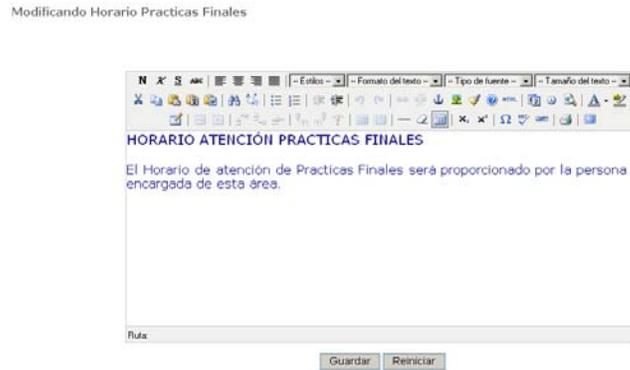
Luego de realizar los cambios necesarios solo queda presionar en cualquiera de las opciones de los botones, puede ser “Guardar” guarda los cambios en la descripción, y “Reiniciar” que deja los valores por *default* en la base de datos.

4.1.2.15. Modificar horario atención prácticas finales

Se navega por la pestaña “Modificar Horarios de Atención” y luego dar un clic en la opción “Horario Atención Prácticas Finales”.

Luego se mostrara la ventana modificar horario atención prácticas finales en la cual se muestra un editor de texto para la realización de cambios.

Figura 47. **Modificar horario de atención de prácticas finales**



Fuente: elaboración propia.

Luego de realizar los cambios necesarios solo queda presionar en cualquiera de las opciones de los botones, puede ser “Guardar” guarda los cambios en la descripción, y “Reiniciar” que deja los valores por *default* en la base de datos.

4.1.2.16. Subir horario de clases prácticas iniciales

Se navega por la pestaña “Subir Horario de Clases” y luego dar un clic en la opción “Horario Prácticas Iniciales”.

Luego se mostrara la ventana de subir horario practicas iniciales en la cual se muestra una casilla en la cual se puede poner el direccionamiento del archivo a subir ó se puede dar clic en examinar para navegar por el equipo en el cual se encuentre el archivo.

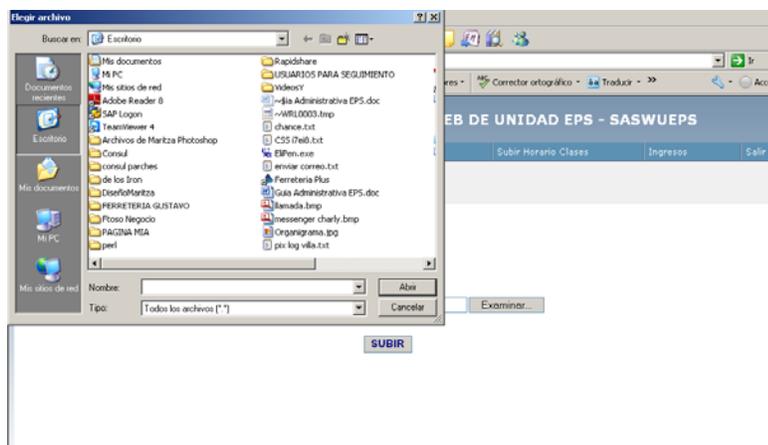
Figura 48. **Subir horario de cursos de prácticas iniciales**



Fuente: elaboración propia.

Luego de dar clic en examinar se mostrara una ventana en la cual se puede buscar el archivo de “Horario de Clases de Prácticas Iniciales” para poder subirlo.

Figura 49. **Seleccionando y subiendo horario de atención de prácticas iniciales**



Fuente: elaboración propia.

Tomar en cuenta que el archivo debe ser formato de *Excel*, sin ningún formato inserto. Luego de seleccionar el archivo se da clic en subir para finalizar.

4.1.2.17. Subir horario de clases prácticas intermedias

Se navega por la pestaña “Subir Horario de Clases” y luego dar un clic en la opción “Horario Prácticas Intermedias”.

Luego se mostrara la ventana de subir horario prácticas intermedias en la cual se muestra una casilla en la cual se puede poner el direccionamiento del archivo a subir ó se puede dar clic en examinar y buscar el archivo.

Figura 50. Subir horario de clases de prácticas intermedias

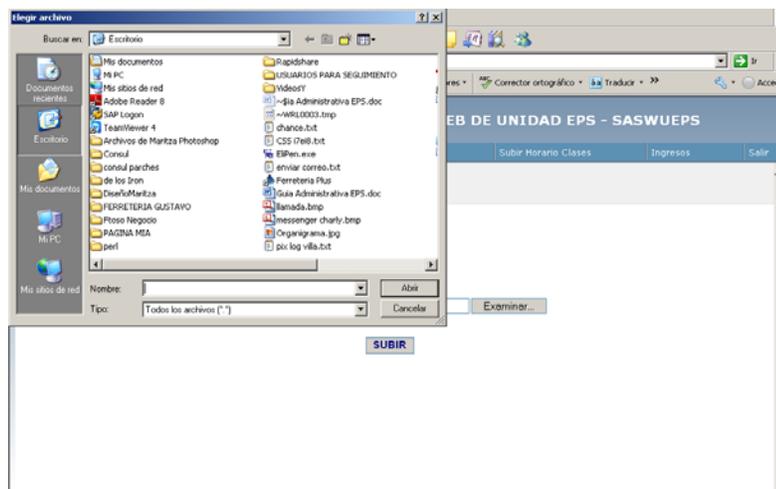


The screenshot shows a web application interface for 'SISTEMA DE ADMINISTRACIÓN DE SITIO WEB DE UNIDAD EPS - SASWUEPS'. The top navigation bar includes links for 'Área Administrativa', 'Acerca de EPS', 'Modificar Horarios de Atención', 'Subir Horario Clases', 'Ingresos', and 'Salir'. The main content area is titled 'Horario Prácticas Intermedias' and contains a form with the label 'Elija un Archivo'. Below the label is a text input field, an 'Examinar' button, and a 'SUBIR' button.

Fuente: elaboración propia.

Luego de dar clic en examinar se mostrara una ventana en la cual se puede buscar el archivo de “Horario de Clases de Prácticas Intermedias” para poder subirlo.

Figura 51. **Seleccionando y subiendo horario de clases de prácticas intermedias**



Fuente: elaboración propia.

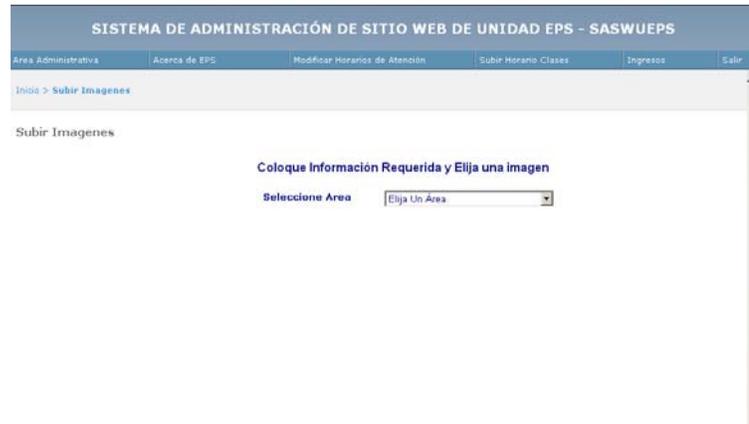
Se debe tener en cuenta que el archivo a subir debe ser en formato de Excel, sin ningún formato inserto. Luego de seleccionar el archivo se da clic en subir y con esto ya se ha concluido de subir el archivo.

4.1.2.18. Ingreso de imágenes

Se navega por la pestaña de ingresos y dar clic en la opción ingreso imágenes.

Luego de esto se ingresará a la ventana de subir imágenes, en dicha ventana aparecerá un menú en el cual se debe de elegir en qué tipo de área se subirá la imagen, las áreas posibles son, prácticas de ingeniería, EPS y DEPTO, según la elección así serán los requerimientos necesarios para subir la imagen.

Figura 52. Subir imágenes



Fuente: elaboración propia.

Suponiendo que se elije subir imágenes de “Prácticas de Ingeniería”, será necesario elegir que sub área pertenece la imagen en la cual se tienen tres opciones, practica inicial, practica intermedia y practica final.

Figura 53. Subir detalles de imagen 1



Fuente: elaboración propia.

En cualquiera de las tres opciones que se elija se solicitará otro nuevo menú en el cual se debe de seleccionar a que escuela pertenece la imagen.

Figura 54. **Subir detalles de imagen 2**



Fuente: elaboración propia.

Luego se solicita indicar la ubicación de la misma, así como el formato que llevará si es horizontal o vertical esto con el fin de mostrar la imágenes correctamente.

Figura 55. **Seleccionando formato y posición de imagen a subir**



Fuente: elaboración propia.

Luego de haber subido la imagen se mostrará la opción de realizar una vista previa de la imagen subida. Con esto se ha terminado de subir la imagen.

Figura 56. Vista previa de imagen



Fuente: elaboración propia.

4.1.2.19. Ingreso de avisos

Se navega por la pestaña de ingresos y dar clic en la opción ingreso avisos.

Luego de esto se ingresará a la ventana de ingreso de avisos, en dicha ventana aparecerá un menú en el cual se debe de elegir en qué tipo de área se subirá el aviso, las áreas posibles son, prácticas de ingeniería, EPS y DEPTO, según la elección así serán los requerimientos necesarios para subir dicho aviso.

Figura 57. Subir avisos de EPS



Fuente: elaboración propia.

Suponiendo que se elije subir avisos de “Prácticas de Ingeniería”, será necesario elegir a que sub área pertenece la imagen en la cual se tienen tres opciones, práctica inicial, práctica intermedia y practica final.

Figura 58. Subir detalles de aviso de EPS 1

SISTEMA DE ADMINISTRACIÓN DE SITIO WEB DE UNIDAD EPS - SASWUEPS

Inicio > SUBIR AVISO

SUBIR AVISOS DE EPS

LLENE LA INFORMACIÓN REQUERIDA

SELECCIONE AREA [ELIJA AREA]

Fuente: elaboración propia.

En cualquiera de las tres opciones que se elija se solicitará otro nuevo menú en el cual se debe de seleccionar a que escuela pertenece el aviso.

Figura 59. Subir detalles de aviso de EPS 2

SISTEMA DE ADMINISTRACIÓN DE SITIO WEB DE UNIDAD EPS - SASWUEPS

Inicio > SUBIR AVISO

SUBIR AVISOS DE EPS

LLENE LA INFORMACIÓN REQUERIDA

SELECCIONE AREA [Practicas De Ingenieria]

SELECCIONE SUBAREA [Practica Inicial]

ELIJA UNA ESCUELA [Elija Una Escuela]

- Escuela De Ciencias y Sistem
- Escuela Industrial y Mecanic
- Escuela Civil
- Escuela De Ciencias Quimica
- Licenciatura en Matematica
- Escuela Elctrica, Electronic
- Licenciatura en Fisica

Fuente: elaboración propia.

Luego de haber elegido la escuela a la que pertenece el aviso, se solicitará indicar un título para el aviso y también indicar a que se refiere dicho aviso. Luego se presiona el botón grabar para finalizar.

Figura 60. **Subir detalles de aviso de EPS 3**

The screenshot shows a web application interface for 'SISTEMA DE ADMINISTRACIÓN DE SITIO WEB DE UNIDAD EPS - SASWUEPS'. The main menu includes 'Inicio > SUBIR AVISO'. The page title is 'SUBIR AVISOS DE EPS'. Below the title, it says 'LLENE LA INFORMACIÓN REQUERIDA'. The form contains the following fields:

- SELECCIONE AREA: Practicas De Ingeniera
- SELECCIONE SUBAREA: Practica Inicial
- ELIJA UNA ESCUELA: Escuela Industrial y Mecanica
- INDIQUE TITULO DE AVISO: Aviso 1
- INDIQUE SU AVISO: primer aviso

A 'GRABAR' button is located at the bottom of the form.

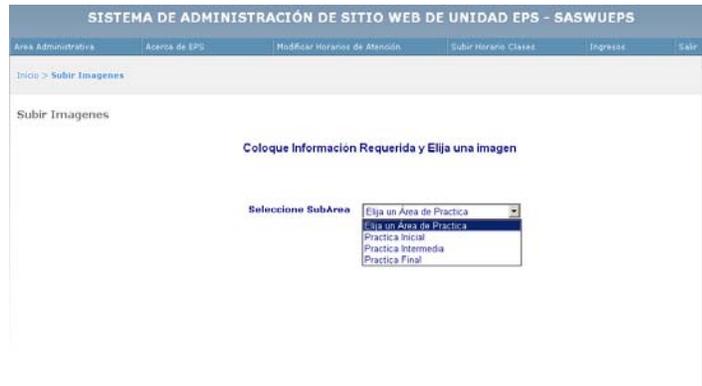
Fuente: elaboración propia.

4.1.2.20. Ingreso de talleres

Se navega por la pestaña de ingresos y dar clic en la opción ingreso talleres.

Luego de esto se ingresará a la ventana de ingreso de talleres, en dicha ventana aparecerá un menú en el cual se debe de elegir en qué tipo de sub área se subirá el taller, las sub áreas posibles son, práctica inicial, practica intermedia y practica final, según la elección así serán los requerimientos necesarios para subir dicho taller.

Figura 61. **Subir talleres**



Fuente: elaboración propia.

Suponiendo que se elije subir taller de “Prácticas Intermedia”, será necesario elegir a qué escuela pertenece el taller en la cual se tienen varias opciones.

Figura 62. **Subir detalles de talleres 1**



Fuente: elaboración propia.

Luego de haber elegido la escuela a la que pertenece el taller, se solicitará indicar la ubicación en la cual se encuentra el taller. Por último se presiona el botón Subir.

Figura 63. **Subir detalles de talleres 2**

SISTEMA DE ADMINISTRACIÓN DE SITIO WEB DE UNIDAD EPS - SASWUEPS

Area Administrativa | Acerca de EPS | Modificar Horarios de Atención | Subir Horario Clases | Ingresos | Salir

Inicio > Subir Imágenes

Subir Imágenes

Coloque Información Requerida y Elija una imagen

Seleccione SubÁrea: Practica Intermedia

Elija Una Escuela: Escuela Civil

C:\Documents and Settings\Administ\Examinar...

SUBIR

Fuente: elaboración propia.

4.1.2.21. **Cambiar la contraseña de usuario administrador**

Estando en la ventana principal de la ventana de administración se presiona clic en el botón cambiar contraseña.

Luego se solicitará una nueva contraseña y la confirmación de la misma. Luego se da clic en cambiar y con esto ya se ha realizado el cambio de contraseña.

Figura 64. **Cambiar contraseña de administrador**

SISTEMA DE ADMINISTRACIÓN DE SITIO WEB DE UNIDAD EPS - SASWUEPS

Area Administrativa | Acerca de EPS | Modificar Horarios de Atención | Subir Horario Clases | Ingresos | Salir

Inicio > Cambio Contraseña Cuenta Administrador

CONTRASEÑA ADMINISTRADOR

Jueves, 10 de Septiembre de 2009 05:38:15 p.m.

Ingrese Una Contraseña Nueva

Nueva Contraseña

Confirme Contraseña

Cambiar

Fuente: elaboración propia.

CONCLUSIONES

1. Es necesario definir y controlar los requerimientos del producto que ha sido elaborado para que sea de calidad, que logre satisfacer y superar las expectativas de los usuarios. Para esto, es necesario controlar a través de inspecciones, control y verificación de avances, el desarrollo del sistema que se está creando y compararlo con los requerimientos que se definieron para la aplicación.
2. Los diagramas facilitan el entendimiento de una aplicación y aportan una conceptualización visual del funcionamiento en cada una de las opciones de los módulos de la aplicación.
3. Es tarea establecer el nivel de seguridad que se implementará para asegurar la información que se encuentre en el sitio. Aunque no exista un esquema de seguridad infalible a los ataques, es importante utilizar estrategias y generar planes de contingencia en caso de sufrir un ataque para neutralizarlo.
4. Es necesario considerar los estándares existentes para el desarrollo de aplicaciones *web* y tratar de apegarse en la medida de lo posible a éstos.
5. Los patrones de arquitectura y diseño son independientes de la plataforma y lenguaje de programación que se utilizará. Dan lineamientos generales que deben considerarse pero no se apegan a un producto en particular. Los patrones de implementación si se apegan a un producto específico.

6. La creación e implementación del sitio *web*, extiende la utilización de los servicios informativos de la biblioteca a muchos más usuarios, apoyando el proceso de aprendizaje de los estudiantes.

RECOMENDACIONES

1. Los ejemplos de patrones sirven como un modelo porque siempre es necesario aplicar el razonamiento y la experiencia en el momento de diseñar una aplicación.
2. Es necesario considerar que cualquier *software* debe estar abierto a un proceso de mejoramiento continuo, en sus diferentes niveles funcionales; orientado a la satisfacción de las necesidades de los usuarios. Por esta razón es importante dar un mantenimiento a las funciones de la aplicación y que éstas se realicen acorde con un modelo que permita llevar un control de los cambios hechos.
3. El mejoramiento del sitio es una responsabilidad que puede delegarse a estudiantes interesados en realizar su práctica supervisada, y cuyos conocimientos y habilidades se puedan poner en práctica.
4. Realizar periódicamente encuestas entre los usuarios para ver que partes del sistema pueden ser mejoradas y para realizar dichas mejoras en la aplicación.
5. Mucha de la información que se encuentre disponible en este sitio, puede ser útil para otras entidades, por esto es vital considerar una arquitectura que deje que la información pueda ser accesada desde otras aplicaciones de forma fácil, y para esto podrían utilizarse servicios *web* que pongan a disposición de estas aplicaciones la información requerida.

6. La administración del ciclo de vida de las aplicaciones *web* es muy importante, ya que es bastante corto y dinámico, por lo que se recomienda utilizar metodologías incrementales e iterativas apoyadas por patrones orientados a roles.

BIBLIOGRAFIA

1. ANÁLISIS *diseño y mantenimiento de software* [en línea]. España: Universidad Nacional de Educación a distancia, 2007. [Consulta: 05 de mayo de 2009]. Universidad Nacional de Educación a Distancia. Disponible en Web: <http://www.ia.uned.es/asignaturas/adms>.
2. AYALA, Wilson. *Arquitectura Cliente/Servidor* [en línea]. Colombia: Blog wilsonayala, 2009. [Consulta: 07 de noviembre de 2009]. *WebBlog* wilsonayala. Disponible en Web: <http://wilsonayala.blogspot.com>.
3. BUTACARA M., César Julio. *Arquitectura .NET* [en línea]. Colombia: Arquitectura de proyectos *web*, 2008. [Consulta: 14 de febrero de 2009]. Arquitecturas .net y proyectos *web*. Disponible en Web: <http://www.monografias.com/trabajos-pdf4/arquitectura-net/arquitectura-net.pdf>.
4. FERNANDEZ, Paola y MARTINEZ, Juan. *Clasificación de las redes* [en línea]. México: Blog payolis and nirvana John, 2010. [Consulta: 05 de enero de 2010]. *WebBlog* payolis and nirvana John. Disponible en Web: http://www.payojohn.blogspot.com/2010_05_01_archive.html.
5. COELLO Costa, Helkyn R. *Como desarrollar software y no morir en el intento* [en línea]. España: Modelos de gestión, desarrollo y alineamiento estratégico, 2008. [Consulta: 25 de octubre de 2009]. Desarrollo profesional de Software. Disponible en Web: http://www.informatizate.net/index.php?option=com_content&view=article

&id=67:como-desarrollar-software-y-no-morir-en-el-intento&catid=36:Art%C3%ADculos%20de%20Miembros%20de%20informatzate&Itemid=62.

6. DOTNETNUKE Corp. *Contenido de Administración de Sistemas* [en línea]. Canadá: Dotnetnuke Corporation, 2002. [Consulta: 21 de septiembre de 2009]. Administración de Sistemas de Información. Disponible en Web: <http://www.dotnetnuke.com/Resources/ResourcesOverview/tabid/2010/Default.aspx>.
7. DYNAMIC Drive. *Menús móviles* [en línea]. USA: Dynamic Drive, 2006. [Consulta: 16 de octubre de 2009]. Persianas desplegadas. Disponible en Web: <http://www.dynamicdrive.com/dynamicindex2/crosstick.htm>.
8. INSTITUTO Nacional de Estadística e Informática. *Tecnología cliente servidor* [en línea]. Perú: INEI, 2007. [Consulta: 03 de noviembre de 2009]. Instituto Nacional de Estadística e Informática de Perú. Disponible en Web: <http://www.inei.gob.pe/biblioinei/pub/bancopub/inf/Lib5038/indice.HTM>.
9. LAFUENTE, Diego Martín. *25 errores comunes en desarrollos web* [en línea]. Argentina: *WeBlog* de tecnología y diseño, 2003. [Consulta: 24 de marzo de 2010]. Programación Profesional. Disponible en Web: http://www.webtaller.com/maletin/articulos/25_errores_comunes_en_los_de_sarrollos_web_2.php.
10. MICROSOFT Corporation. *Arquitectura de aplicaciones web: Diseño de aplicaciones y servicios* [en línea]. USA: MSDN en español, 2006. [Consulta: 23 de junio de 2010]. Microsoft Corporation. Disponible en Web: <http://msdn.microsoft.com/es-es/library/ms954595.aspx>.

11. MICROSOFT Corporation. *Introducción a la administración del ciclo de vida de las aplicaciones con Visual Studio* [en línea]. USA: MSDN en español, 2009. [Consulta: 15 de febrero de 2010]. Microsoft Corporation. Disponible en Web: <http://msdn.microsoft.com/library/dd286491.aspx>.

12. SÁNCHEZ, Carlos. *Un proyecto open source para el negocio textil mayorista desarrollado con tecnologías open source innovadoras* [en línea]. España: ONess, Universidad de Coruña, 2004. [Consulta: 27 de octubre de 2009]. Universidad de Coruña España. Disponible en Web: <http://oness.sourceforge.net/proyecto/html/index.html>.

