



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

CAMPUS ARAGÓN

**“SISTEMA DE AVISOS A TRAVÉS DE PAGINA WEB
EN EL SERVIDOR DE CORREO ELECTRÓNICO
DEL INSTITUTO DE INGENIERÍA”**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A:

LUIS ALBERTO BARRÓN CEDEÑO

ASESOR: M. EN C. JESÚS DÍAZ BARRIGA ARCEO



MÉXICO

2004



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*Al la única que hasta ahora me sigue soportando
y que por cierto, estuvo ahí cuando nací*

Al todos y a nadie



Índice de contenido

Introducción	3
Capítulo 1. Comunicación interna en el Instituto de Ingeniería. Presente y futuro.	
1.1 El Instituto de Ingeniería	6
1.2 La Coordinación de Sistemas de Cómputo, misión y características	9
1.3 Sistema de Avisos en el servidor de correo electrónico “Pumas”	12
1.4 Nuevas necesidades de comunicación	15
1.5 Análisis de soluciones existentes	17
1.5.1 Aliases	17
1.5.2 Gestores de listas	18
Capítulo 2. Marco tecnológico	
2.1 Ciclo de vida	21
2.2 Metodologías de desarrollo	23
2.2.1 Metodología Yourdon	23
2.2.2 Proceso Unificado	24
2.2.3 Extreme Programming	26
2.3 Sistemas orientados a web	30
2.4 Editores de HTML	31
2.5 Plataformas para desarrollo web	33
2.5.1 .net	34
2.5.1.1 Servidor IIS	35
2.5.1.2 SQL Server	35
2.5.1.3 ASP	35
2.5.2 LAMP	36
2.5.2.1 Linux	36
2.5.2.2 Servidor Apache	37
2.5.2.3 MySQL	37
2.5.2.4 PHP	38

2.5.3	Algunas opciones independientes de plataforma	40
2.5.3.1	Oracle	40
2.5.3.2	JAVA	40
2.5.4	Decisión de la plataforma	42

Capítulo 3. Desarrollo de la aplicación.

3.1	Entrevistas con los usuarios	48
3.1.1	Reunión con el jefe del Área de Servidores UNIX	48
3.1.2	Reunión con el Coordinador de Sistemas de Cómputo	50
3.1.2.1	Primera reunión	50
3.1.2.2	Segunda reunión	52
3.1.3	Conclusiones de las reuniones	53
3.2	Diseño de la interfaz	54
3.2.1	Separación de etapas de la aplicación	54
3.2.2	Desarrollo de la interfaz	56
3.3	Base de datos	59
3.3.1	Análisis de datos	59
3.3.2	Diseño de las tablas	60
3.3.3	Implementación de la base	61
3.4	Programación de módulos	63
3.4.1	Autenticación	63
3.4.2	Adaptación de código existente	65
3.4.2.1	Editor	65
3.4.2.2	Envío del mensaje	67
3.4.3	Detección / Corrección de errores	68

Capítulo 4. Implementación y liberación.

4.1	Pruebas internas	74
4.1.1	Pruebas de formato	74
4.1.2	Pruebas de seguridad	75
4.1.3	Pruebas de carga (desempeño)	76

4.1.3	Pruebas de carga (desempeño)	76
4.1.4	Pruebas de contingencia	77
4.2	Implementación	78
4.2.1	Etapas de liberación	78
4.2.2	Capacitación	78
4.3	Modificaciones subsecuentes	80
	Conclusiones	83
	Apéndice I. Manual de Usuario	85
	Referencia Bibliográfica	99

Introducción

Las necesidades de comunicación en el ámbito humano han cambiado radicalmente en la última década. Hemos pasado de un escenario de telecomunicación dominado prácticamente por la telefonía fija a una combinación entre telefonía fija y portátil y un actor completamente nuevo: el correo electrónico. Resulta común incluso que las personas basen sus relaciones únicamente en el correo electrónico.

Paulatinamente, las personas han adoptado este medio no solo para relacionarse socialmente, en la actualidad buena parte de las operaciones en las empresas e instituciones dependen directa o indirectamente de este recurso que día a día se vuelve más imprescindible.

El proyecto planteado en el presente trabajo, trata de la explotación del correo electrónico con un fin específico, la distribución de información de carácter institucional. Si bien, el sistema puede ser implementado bajo diversas circunstancias y en distintos lugares, la institución de la que hablamos en este caso en particular es el Instituto de Ingeniería de la Universidad Nacional Autónoma de México ubicado en la Ciudad Universitaria.

Se trata de un sistema para el envío de mensajes de carácter informativo a través del correo electrónico. Si bien, sería inevitable que el sistema propuesto pudiera ser utilizado para objetivos que pudieran ser considerados inapropiados como el conocido como correo "spam", se trata de una aplicación estrictamente de uso interno y bajo condiciones y políticas apropiadas que evitan el uso incorrecto del sistema.

En el primer capítulo, se observa una breve reseña de lo que es el Instituto de Ingeniería en aspectos como estructura, distribución y principales actividades. Luego, se abordará el problema con un análisis sobre la situación actual del Instituto en aspectos de comunicación interna.

Conociendo ya el entorno y las necesidades de comunicación del Instituto, en el segundo capítulo se encuentra el marco tecnológico que permitirá seleccionar los recursos y metodologías que serán utilizados en el análisis, desarrollo e implementación del sistema para obtener los mejores resultados.

En el tercer capítulo, se describen las distintas etapas del desarrollo de la aplicación, desde las entrevistas con los usuarios, análisis y diseño hasta el desarrollo e implementación del sistema.

Para la liberación del sistema, es necesaria la realización de diversas pruebas. En el cuarto capítulo, se describen las pruebas realizadas al sistema previo a su liberación y después de ella además de la capacitación y las modificaciones implementadas al sistema debido a su solicitud por parte de los usuarios después de comenzar a utilizarlo.

Cabe señalar, que la aplicación desarrollada, no está cerrada al uso por parte del Instituto de Ingeniería, de hecho durante su concepción y desarrollo se ha planteado su adaptación a otras instituciones tanto públicas como privadas.

Incluso, es del interés del autor que este sistema sea implementado en la Escuela Nacional de Estudios Profesionales Aragón de la UNAM para la comunicación entre el personal administrativo y académico y como una herramienta de difusión de eventos académicos, culturales, deportivos e incluso administrativos (inscripciones, materias, etc.) con toda la comunidad aragonesa (incluyendo estudiantes).

CAPÍTULO 1

Comunicación interna en el Instituto de Ingeniería
Presente y futuro.

1.1 El Instituto de Ingeniería

El Instituto de Ingeniería de la Universidad Nacional Autónoma de México, fue fundado en 1956 (por lo que está a punto de cumplir sus primeros 50 años). Forma parte del Subsistema de Investigación Científica de la Universidad Nacional Autónoma de México y orgánicamente se encuentra dentro de la Coordinación de la Investigación Científica.

Sus principales objetivos son mejorar los conocimientos, métodos y criterios en ingeniería y contribuir a la formación de expertos en esta rama, así como promover la más alta calidad en la práctica profesional.

Las actividades que se llevan a cabo en el Instituto son: investigación técnica y aplicada, apoyo al desarrollo tecnológico y análisis de los requerimientos sociales cuya solución puede ser aportada por la ingeniería. Así mismo, proporciona servicios de ingeniería a los diversos sectores de la sociedad con el propósito de contribuir al avance de los objetivos propios de la Universidad.

Si bien, no existe una métrica establecida ni algún organismo que lo avale, el Instituto de Ingeniería es considerado en los hechos el mayor generador de investigación a nivel nacional en las ramas de la Ingeniería Civil (Sismología, Geotecnia, Estructuras) y Ambiental (Bioprocesos, Hidráulica)

El Instituto de Ingeniería está conformado por 12 edificios y la Torre de Ingeniería. Se subdivide en tres subdirecciones que son Ambiental, Estructuras y Electromecánica. Éstas a su vez acogen en conjunto 15 áreas de investigación que son detalladas en el siguiente cuadro:

Subdirección de Ambiental

- **Bioprocesos Ambientales** Realiza investigación y desarrollo de procesos biotecnológicos ambientales, y su transferencia al sector privado. Es líder nacional en tratamiento anaerobio de aguas residuales.
- **Hidráulica** Estudia el comportamiento y diseño de estructuras hidráulicas, hidrología, hidráulica fluvial y marítima, estabilidad de sistemas hidromecánicos y optimización.
- **Ingeniería Ambiental** Se orienta a la investigación y desarrollo de sistemas de tratamiento y potabilización de aguas residuales, convencionales y problemáticas.
- **Ingeniería de Procesos** Desarrolla y aplica métodos y técnicas avanzadas en ingeniería de termo-fluidos y de seguridad, procura la protección del ambiente y los trabajadores de la industria de transformación.

Subdirección de Electromecánica

- **Automatización** Implementa esquemas de control y supervisión de procesos industriales basados en circuitería electrónica, programación y desarrollo de algoritmos.
- **Ingeniería Mec. Térmica y Fluidos** Estudia y diseña mecanismos industriales para la automatización de procesos, dispositivos para absorción de energía sísmica en edificios, máquinas rotatorias, compresores y engranes.
- **Ingeniería de Sistemas** Investiga metodologías y realiza estudios en planeación urbana y regional, desarrollo rural e industrial, innovación tecnológica y planeación. Aborda aspectos de inteligencia artificial
- **Instrumentación** Realiza investigación en electrónica e instrumentación aplicada. Proporciona asesoría en la selección de equipos de medición y mantenimiento de aparatos de laboratorio.
- **Sistemas de Cómputo** Se trata en general de un grupo que hace planeación e implementación de la infraestructura de cómputo al interior del Instituto de Ingeniería¹

Subdirección de Estructuras

- **Estructuras y Materiales** Realiza estudios analíticos y experimentales sobre el comportamiento de estructuras y materiales expuestos a diversas acciones que afectan la vida útil de las construcciones.
- **Geotecnia** Investiga la estructura y comportamiento mecánico de suelos naturales y compactados, finos y granulares y enrocamientos.
- **Ingeniería Sismológica** Reúne a investigadores especializados en la observación, análisis y modelado de terremotos para reducir el riesgo sísmico en los asentamientos humanos.
- **Mecánica Aplicada** Efectúa investigaciones teóricas y aplicadas en análisis de respuesta inelástica de edificios ante eventos sísmicos. Determina el riesgo sísmico en sitios específicos.
- **Sismología e Ins. Sísmica** Investiga y desarrolla proyectos para el registro de temblores en zonas de alto riesgo sísmico o en núcleos de población que puedan ser afectados.
- **Vías Terrestres** Estudia sistemas de transportes en relación con el diseño geométrico y estructural, operación, conservación y reconstrucción de carreteras y pistas aeroportuarias.

Tabla 1 Organización interna del Instituto de Ingeniería

¹ Más adelante, será abordada con mayor detalle.

La comunidad que conforma al Instituto de Ingeniería es de aproximadamente 1200 personas, que incluye investigadores, becarios a nivel licenciatura, maestría y doctorado, técnicos académicos, personal administrativo y de servicio.

Los principales patrocinadores del Instituto de Ingeniería son:

DGAPA, Fundación Politécnico, Comunidad Económica Europea, Consejo Nacional de Ciencia y Tecnología, Dirección General de Construcción y Operación Hidráulica, Acatex, Atlántida del Sur, E&M Inc., Ferrocarriles Nacionales de México, Fundación México-Estados Unidos para la Ciencia, Ibtch, PEMEX, Instituto Mexicano de Tecnología del Agua, Banco Interamericano de Desarrollo, Comisión Federal de Electricidad, Comisión Nacional del Agua, Gobierno del Distrito Federal, CENAPRED, SCT, Compañía de Luz y Fuerza, Universidad Autónoma de Puebla, Universidad de California.

1.2 La Coordinación de Sistemas de Cómputo, misión y características.

La Coordinación de Sistemas de Cómputo del Instituto de Ingeniería (CSC-II) tiene como función básica la administración de los recursos de cómputo al interior del Instituto. Para ello, cuenta con 28 personas divididas en cinco áreas y un coordinador.

Su objetivo primordial es dar soporte a las investigaciones que se realizan en las demás coordinaciones. Cabe señalar, que se trata de la única coordinación del II² que no genera ninguna investigación ni proporciona servicio alguno a entidades externas. Las seis áreas que componen a la CSC y sus actividades son:

- ♦ *Bases de datos*

Cuenta con nueve integrantes. Desarrolla y mantiene los múltiples sistemas de información del II como la base de datos académica, de becarios y la base de la Secretaría Académica entre otros sistemas de objetivo específico para satisfacer necesidades tanto de carácter administrativo como académico. Entre los manejadores de bases de datos (DBA) que utilizan podemos mencionar SQL Server y Sybase. Los lenguajes de programación de más común uso son ASP y Visual Basic.

- ♦ *Desarrollo WEB*

Solo cuenta con una persona. Se encarga del diseño gráfico e implementación de las páginas electrónicas de la comunidad del II. Diseña logos de diversos congresos que organiza o de los que es partícipe el II. También lleva a cabo la actualización de información de las páginas disponibles en los distintos servidores.

- ♦ *PC's y periféricos*

Con siete integrantes, se encarga de realizar trabajos de mantenimiento a los equipos de cómputo del personal del II. Sus funciones comprenden la instalación y configuración de sistemas operativos y paquetería en general (sobre sistema operativo Windows) además de la solución de problemas relacionados.

- ♦ *Redes y conectividad*

Sus cuatro integrantes se encargan de realizar la planeación, instalación y administración de la red local del II (REDII), además de mantener la seguridad lógica y física de la misma. En la actualidad

2 En adelante debe interpretarse II como Instituto de Ingeniería

su principal proyecto consiste en portar la red de ATM a Gigabit. Su siguiente proyecto es la migración a telefonía basada en voz sobre IP.

♦ *Servidores Windows NT*

Cuenta con 4 integrantes. Administra los servidores Windows NT, incluyendo cuentas de usuarios, mantenimiento del servicio de DHCP³ y administración del servidor de correo electrónico para becarios.

♦ *Servidores UNIX*

Sus características y funciones son:

- ♦ Administración del servidor de página web del II y de correo electrónico del personal académico y administrativo **pumas.iingen.unam.mx**
- ♦ Administración del servidor para cálculo numérico de alto rendimiento **ollin**
- ♦ Administración del servidor para cliente de correo por página web **merlin**
- ♦ Administración del servidor para ftp e instalaciones de sistema operativo Linux: **linux**
- ♦ Desarrollo de aplicaciones utilizando software libre.
- ♦ Administración de estaciones de trabajo pertenecientes a la Coordinación de Sismología⁴
- ♦ Administración de estaciones de trabajo Linux de todo el II.

Los puestos de que se compone el Área de Servidores UNIX son:

3 Dynamic Host Configuration Protocol. Protocolo empleado para que los clientes en una red obtengan su configuración de forma dinámica a través de un servidor. La configuración puede incluir dirección IP, máscara de red y dirección de *broadcast* entre otros. Permite acelerar la configuración de muchos *hosts* en una red evitando en gran medida los posibles errores humanos.

4 Utilizadas para simulación de eventos sísmicos y compilación de programas en lenguaje C y Fortran.

Puesto	Funciones principales
Jefe de área	Coordinación de todas las actividades. Administrador de los servidores <i>pumas</i> y <i>ollin</i> además de estaciones de trabajo Solaris y SuSE Linux.
Administrador de servidores y ws linux	Administración de los servidores <i>linux</i> y <i>merlin</i> además de estaciones de trabajo con SO Linux
Administrador de servidores y desarrollo de utilerías	Administración del servidor <i>pumas</i> , desarrollo de aplicaciones de automatización de procesos, atención al personal.
Encargado de seguridad	Investigación e implementación de herramientas de seguridad. Administración del servidor <i>ayax</i> .

Tabla 2 Integrantes del Área de Servidores UNIX

El Área de Servidores UNIX cuenta con el siguiente equipo:

Equipo	Sistema Operativo	Función
Servidor SUN Enterprise 250	Solaris 8	Servidor de página web, FTP y correo electrónico.
Servidor Intel SRKA4	Red Hat Linux 8	Nodo principal del cluster <i>ollin</i> , En él se compilan y ejecutan programas en lenguaje Fortran y C principalmente.
2 Servidores Intel PC	SuSE Linux 9.1	Servidor de acceso a correo electrónico por página web y servidor de instalaciones Linux.
4 estaciones Intel PC	SuSE Linux 9.1	Estaciones de trabajo del personal del área.
2 estaciones SGI O2	IRIX 6.5	Sin función actualmente. Se planea utilizarlas para realizar pruebas de clusters en SGI y generación de animaciones.
Servidor SUN Enterprise 250	Solaris 8	Dedicada al Grupo de Ingeniería Lingüística. Proporciona servicios de Apache, Postgres, MySQL, PHP y Python.

Tabla 3 Equipo del Área de Servidores UNIX

1.3 Sistema de Avisos en el servidor de correo electrónico "Pumas".

La comunicación entre el personal del II es crucial para la correcta operación del mismo. Toda la información que debe ser difundida entre la población completa o grupos específicos, requiere de un mecanismo efectivo que garantice que toda persona interesada en un mensaje lo reciba.

Los medios utilizados para ésta comunicación son muy diversos e incluyen los carteles, oficios, vía oral, etc. Sin embargo, los tres tienen algunos inconvenientes, los carteles y oficios pueden ser costosos, tardan en ser impresos y alguien tiene que ocuparse de colocarlos. Para una comunicación oral, ya sea en persona o por vía telefónica, se requiere que un grupo de personas se encargue de distribuir la información de manera personalizada; es demasiado el tiempo que se debe invertir y el problema aumenta si no se encuentra a la persona que debe ser informada.

Supongamos que se requiere que los becarios de todas las coordinaciones del II, proporcionen ciertos papeles (digamos su historial académico) a la Secretaría Académica. Parece que cualquiera de los tres métodos tratados antes podría funcionar, pero tomemos algo en cuenta: el II cuenta con más de 700 becarios entre estudiantes de licenciatura, maestría y doctorado, estos se encuentran dispersos en 12 edificios y la Torre de Ingeniería. Por si fuera poco, en la mayoría de las ocasiones solo el tutor de cada becario conoce su horario de trabajo al interior del Instituto.

Por las razones planteadas, ni con un oficio, ni con carteles, ni por vía oral se logrará que un porcentaje significativo de becarios haga llegar su historial a tiempo.

Ahora bien, tomemos en cuenta que todo becario, académico o administrativo del II posee una cuenta de correo electrónico proporcionado por el mismo Instituto⁵, la solución salta a la vista. Difundir los anuncios por este medio tiene muchas ventajas sobre las otras opciones, entre ellas podemos considerar:

- El mensaje debe ser escrito una sola vez y enviado a todas las personas pertinentes.
- No es necesario que el destinatario se encuentre al momento de ser enviado el mensaje (como ocurriría por vía telefónica o de manera personal).
- Existe una prueba de la difusión de la información, todo es guardado en una bitácora y ante alguna confusión o reclamo, se cuenta con registro del envío incluyendo a quienes llegó.

No obstante, como se detallará más adelante, estas ventajas aún no son suficientes.

En 1998, comenzó a efectuarse la difusión de información de interés a través del correo electrónico, sin embargo, aún existía un inconveniente, una persona necesitaba tener una lista con las personas que debían recibir el mensaje y aunque solo requería escribirlo una vez, debía enviarlo de manera individual tantas veces como personas estaban incluidas en la lista. Si el mensaje estaba

⁵ Ya sea *usuario@pumas.iingen.unam.mx* si es académico o administrativo o *usuario@iingen.unam.mx* si se trata de un becario.

dirigido a todo el personal, tal vez tardaría un día completo en enviarlo a todos.

Por ello se vio la necesidad de crear un programa de envío masivo de correo, que para 1999 fue denominado INFO⁶.

INFO fue escrito en Shell Script⁷; su principio básico era la creación previa de archivos que contuvieran un listado de direcciones de correo electrónico para que el sistema lo leyera y enviara el correo de manera autónoma con el menor esfuerzo posible por parte del personal.

Veamos brevemente cómo funciona INFO.

Para la discriminación de personal al que será enviado el mensaje, se tomó como factor al tipo de personal con los siguientes archivos de lista:

<i>Archivo</i>	<i>Personal</i>
ACA	Investigadores y técnicos académicos.
ADM	Personal administrativo (secretarías, funcionarios, etc)
BEC	Becarios de nivel licenciatura, maestría y doctorado
ACA-BEC	Suma de ACA y BEC
ACA-ADM	Suma de ACA y ADM
II	Suma de ACA, BEC y ADM

Tabla 4 División de personal en base a categoría

Cada uno de estos archivos tiene el siguiente formato:

usuario@pumas.iingen.unam.mx:Nombre del Usuario, Coordinación, Categoría

De esta manera, el sistema toma cada línea como entrada y va tomando los mensajes uno por uno para enviarlos de manera personalizada.

El mensaje es enviado a una cuenta en particular del sistema (que en realidad no existe) y

6 Realizado el jefe del Área de Servidores UNIX.

7 Específicamente Csh.

por medio del archivo *etc/aliases*, se envía el mensaje como entrada al script encargado de realizar todo el proceso.

Analicemos ahora el procedimiento que se debe seguir para enviar un mensaje.

1. La persona que requiere el envío de algún mensaje lo hace llegar al coordinador de Sistemas de Cómputo o a cualquiera de los integrantes del Área de Servidores UNIX.
2. Se requiere la autorización del coordinador para proceder al envío.
3. El mensaje es enviado a través de algún cliente de correo como pine, Eudora o Outlook.

Es necesario estructurar el asunto de la siguiente manera:

Asunto : Lista : Clave

Con el sistema INFO, se logró una enorme mejora en la difusión de información. Si bien nunca ha suplantado otros medios de comunicación (ni fue uno de sus objetivos) cumplió con satisfacer una necesidad importante del II.

Sin embargo, como veremos en el siguiente apartado, en la actualidad se requiere de algo más. Las necesidades han aumentado e INFO ha llegado a su límite de crecimiento.

1.4 Nuevas necesidades de comunicación.

Si bien INFO es un sistema que cumple totalmente con su objetivo, existen algunos inconvenientes que con el tiempo se han vuelto críticos. Veamos un resumen de ellos.

- x Si no está presente ninguna de las personas encargadas del envío de avisos, se debe esperar hasta que alguno llegue, el aviso se queda parado. Este hecho se hace todavía más frustrante cuando estando presente una persona del Área de Servidores UNIX, lo único que falta es la autorización del coordinador que por alguna razón no está disponible.
- x El correo electrónico en texto plano limita la comunicación. Es necesario usar texto con formato, imágenes e incluso fondos.

Tal vez los peores casos son los tratados a continuación, veamos por qué.

- x Un mensaje solo discrimina a sus destinatarios en base al tipo de personal de que se trata (ACA, ADM o BEC), sin embargo, hay información de interés solo para ciertas coordinaciones. Bajo éste sistema, si se requiere la atención de académicos de la Coordinación de Sistemas, el mensaje necesariamente debe ser enviado a los académicos de todo el II. Para el 90% de los destinatarios, el mensaje no es de interés.
- x Para el envío de un aviso, necesariamente se debe pasar por la Coordinación de Sistemas de Cómputo (análogo a demasiada burocracia), parece un puente demasiado largo. Si el director o algún subdirector requiere del envío de un aviso, solo lo puede hacer a través de esta área, implica tardanza en la comunicación y mayor responsabilidad para el personal de Cómputo.

Uno de estos inconvenientes ha sido solucionado con la creación de INFO2. Este es capaz de utilizar un cliente de correo como Outlook para la edición de mensajes, incluyendo el formato MIME⁸ (Multipurpose Internet Mail Extension), que "permite integrar dentro de un mensaje de correo electrónico archivos binarios (como imágenes) además de aceptar 'rich text' que incluye colores, tamaños de fuente y alineación de textos"⁹.

Sin embargo esto no es suficiente. Es necesario dar un nuevo salto. Viejas y nuevas necesidades deben ser satisfechas. Entre las primordiales podemos mencionar:

- Eliminar la dependencia del Sistema de Avisos respecto al personal de la CSC.
- Proporcionar una interfaz sencilla de usar y amable al usuario.
- Eliminar la necesidad de uso de un cliente de correo determinado.
- Permitir el envío de avisos sin necesidad de encontrarse dentro de las instalaciones del II.
- Delimitar mejor la discriminación de destinatarios incluyendo no solo su categoría dentro del II,

⁸ MIME trabaja al adjuntar un encabezado a cada archivo. El encabezado incluye el tipo de contenido y el subtipo de la información que contiene el archivo. Esto permite que el servidor y el navegador sirvan y presenten correctamente cada archivo. También permite a los usuarios crear páginas que combinen archivos con distintos tipos de información.

⁹ Lerner, M. <http://www.learnthenet.com/spanish/glossary/mime.htm>

sino la coordinación a la que pertenece, el puesto que desempeña e incluso el edificio en que se encuentra su lugar de trabajo.

Por ello, es necesario decidir qué se debe hacer, un sistema basado en INFO e INFO2, otro sistema basado en las mismas herramientas (SH + cliente de correo) o un sistema nuevo con un enfoque de uso y creación totalmente distinto.

1.5 Análisis de soluciones existentes.

En el desarrollo de aplicaciones y en algunos otros ámbitos, existe una regla no escrita que es conveniente seguir “sí ya existe no lo hagas”. Antes de comenzar a desarrollar un nuevo sistema, es importante asegurarse de que no existe ya algo que haga lo mismo. Se pueden ahorrar muchas horas de trabajo y desgaste físico.

Para lo que requerimos, existen ya aplicaciones similares que podrían satisfacer las necesidades de comunicación en el II. Veamos algunas de ellas.

1.5.1 Aliases

Sendmail cuenta con un archivo denominado */etc/aliases*¹⁰. El principio básico es escribir un “alias” que no es más que un nombre (identificador) seguido de una dirección de correo, lo que se ve muy parecido a ésto:

```
alberto: alberto@pumas, albarron@gmail.com
```

En éste caso, todos los mensajes que lleguen a la cuenta *alberto* en nuestro servidor, son redirigidos a *alberto@pumas* y a *albarron@gmail.com*. Esta parece ser una buena solución.

Si es necesario hacer una lista, basta con escribir el identificador que tendrá seguido del grupo de direcciones como se observa a continuación:

```
academicos: alberto, juan@litoral, ernesto, amalia@pumas, fernando@pumas, maria@pumas,  
victor@pumas, benjamin@pumas, lucia@unam.mx albar@servidor.unam.mx,  
omar_2202@hotmail.com, nenca@hotmail.com, "/home/alberto/academicos",  
jrafaeln@yahoo.com, include:/home/abc/.listas/academicos
```

```
becarios: luis, juan, ernesto, academicos, atala@yahoo.com, edgarmp1@yahoo.com
```

Del ejemplo anterior, cabe destacar varios aspectos importantes:

- ✓ Si la cuenta de correo se encuentra en el mismo servidor, no es necesario poner *@nombre.dominio*
- ✓ Cuando las direcciones de destino, se encuentran en un servidor bajo el mismo dominio del local, basta con el identificador seguido del nombre de la máquina, *juan@litoral* por ejemplo
- ✓ Solo cuando el destino está en otro dominio, es necesario poner la dirección completa como en *nenca@hotmail.com*
- ✓ No solo se puede redirigir hacia una cuenta de correo, también es posible hacerlo a un archivo en el

¹⁰ En algunos sistemas, es */etc/mail/aliases* y una liga apunta a él desde */etc/aliases*

mismo servidor como en el caso de `/home/alberto/academicos`. Esto permite tener más orden en el manejo de direcciones y del propio alias.

- ✓ Se pueden incluir archivos que contengan varias direcciones de correo, basta con poner *include:* seguido del archivo que se utilizará
- ✓ Los destinatarios pueden ser otro alias.

Alias parece ser la mejor solución (además de ser muy sencilla). Únicamente existen algunos inconvenientes:

- ✗ Cualquiera puede escribir a un alias y hacer llegar sus mensajes a todas las direcciones que se encuentren (lo que se presta a SPAM, bromas, etc), la posibilidad de un mal uso está latente.
- ✗ De manera directa, no se puede llevar un registro de los envíos realizados.

1.5.2 Gestores de listas

Un gestor de listas tiene como objetivo automatizar los procesos de eliminación y adición de direcciones de correo a una lista. Además, es posible limitar el origen del mensaje a únicamente aquellas direcciones que se encuentran en la misma lista.

Uno de los gestores de lista más conocido es Majordomo. Veamos una breve descripción.

“El programa Majordomo tiene como objetivo la automatización en el manejo de listas de correo electrónico. Recibe los comandos para el mantenimiento de la lista también vía correo. Una vez que la lista es generada, virtualmente todas las operaciones pueden ser realizadas de manera remota a través de correo, sin requerir intervención del administrador de correo sobre la lista.

Majordomo controla una lista de direcciones para algún sistema de transporte de correo (como sendmail o qmail). Majordomo no envía por sí mismo correos.”¹¹

Las principales características de Majordomo son:

- ◆ Soporta varios tipos de listas, incluyendo las moderadas¹²
- ◆ Toda la administración es realizada por correo. El propietario de la lista, no requiere acceder al servidor para realizar estas actividades
- ◆ Está diseñado en forma modular.
- ◆ Desarrollado en lenguaje Perl (solo un módulo está realizado en C)

¹¹ Trad. <http://www.greatcircle.com/majordomo/>

¹² En una lista moderada, todos los mensajes son leídos o filtrados por una persona (moderador), que se encarga de revisarlos antes de enviarlos. La principal desventaja es que su mantenimiento conlleva mucho tiempo.

- ◆ Filtros de listas basados en encabezados o expresiones regulares en el cuerpo

El siguiente cuadro, señala las ventajas y desventajas de mayordomo respecto a esta aplicación.

<i>Ventajas</i>	<i>Desventajas</i>
Su licencia permite obtenerlo de manera gratuita, copiarlo y modificarlo, pero solo para uso propio, no puede ser redistribuido.	Se requiere crear una cuenta en el sistema
Existe una comunidad de usuarios dispuesta a la mutua ayuda.	Está más bien enfocado a foros.
Está desarrollado en Perl por lo que se cuenta con los códigos y solo una pequeña parte en C (también es proporcionado el código)	Cualquiera puede escribir un mensaje. Requeriría de un moderador, con lo que volvemos al mismo problema: dependencia.
Trabaja con Sendmail	El envío de adjuntos no es recomendable, pues existe un límite en el tamaño del mensaje enviado que comúnmente es rebasado al incluirlos.
	Limita el envío de correos a direcciones contenidas en la misma lista.

Tabla 5 Ventajas y desventajas de mayordomo.

Existen paquetes parecidos a Majordomo, como Mailman (GNU). Su función es básicamente la misma.

CAPÍTULO 2

Marco tecnológico.

2.1 Ciclo de vida

En el desarrollo de un sistema, se deben tomar en consideración varios aspectos que permitan alcanzar el objetivo de la manera más eficiente. Uno de ellos es su ciclo de vida que consta de la definición de etapas por las que pasará un sistema desde su concepción hasta su puesta en funcionamiento (e incluso tiempo después). Existen varias vertientes de las etapas del ciclo de vida que normalmente solo tienen diferencias sutiles entre sí.

El ciclo de vida de un sistema comprende las etapas:

1. Definición de requisitos
2. Especificación funcional
3. Descripción del diseño
4. Realización de los programas
5. Métodos de prueba
6. Mantenimiento

Previo a la realización del sistema, es necesario hacer un planteamiento del problema y las actividades que se deben realizar. Para saber las características del problema, las necesidades de desarrollo y determinar la viabilidad de la solución, es necesario pasar por tres pasos básicos:

- Formulación del problema.
- Búsqueda y desarrollo de soluciones.
- Evaluación de soluciones.

Veamos un breve esbozo de en qué consistirán las etapas del ciclo de vida de la aplicación propuesta.

1. Definición de requisitos

Se requiere una aplicación independiente del sistema operativo del lado del usuario además de una manera fácilmente comprensible para el envío de mensajes informativos. Es importante que el usuario no tenga acceso a los datos de las personas a las que les envía el correo. Simplemente se debe realizar la selección de personal en base a características comunes como coordinación o categoría al interior del II.

Por el lado de los administradores, es necesario llevar una bitácora con datos como fecha y hora de envío, remitente y destinatarios además de una copia del mensaje y archivos adjuntos. El acceso debe ser restringido y estar disponible dentro y fuera de las instalaciones.

2. Especificación funcional

El sistema debe contar antes que nada con un módulo de autenticación cifrada. Un

mecanismo de selección de destinatarios sencillo y una interfaz de escritura sencilla pero con opciones para texto enriquecido. Por otro lado, debe contar con una confirmación del correcto envío del mensaje a cada una de las direcciones de correo correspondientes.

3. Diseño

Debido a que el diseñador y programador del sistema son una misma persona, los documentos técnicos y descriptivos del sistema pueden ser bastante sencillos. Consistirán en las definiciones de las tablas de la base de datos, las relaciones entre las mismas y las llaves tanto primarias como relacionales¹³. Básicamente, es necesaria una tabla de personal con datos personales como dirección de correo, clave de coordinación, clave de categoría, extensión y cubículo entre otros. Además se necesita una tabla de coordinaciones y una de categorías.

Por la parte de la presentación, se utilizarán colores que se identifiquen como parte del II, el sistema estará dividido en 5 etapas: Autenticación, Selección de destinatarios, Escritura del mensaje, Envío del mensaje y Confirmación.

4. Realización

Con el uso de algún lenguaje estructurado, se realizará el "background" y el "foreground" de la aplicación muy probablemente de manera combinada. Lo mejor sería realizar primero una propuesta general de la presentación del programa, para después realizar el diseño gráfico de cada una de las etapas seguido del desarrollo de los procedimientos correspondientes.

5. Prueba

Las primeras pruebas deben ser realizadas con destinatarios ajenos al personal del II. Para ello, se puede hacer uso de alguno de los servidores de prueba con que cuenta el área además de la inclusión de direcciones de correo comerciales¹⁴. Esta etapa servirá para la prueba simplemente del correcto envío del mensaje (en aspectos de formato y presentación) para luego aplicar pruebas de carga al servidor y comportamiento en caso de contingencias como las ocurridas cuando la dirección no existe o está sobre la cuota permitida por el proveedor.

6. Mantenimiento

Es inevitable que luego de la puesta en funcionamiento se requieran modificaciones desde la inclusión de nuevos registros del personal hasta agregar nuevas funciones a la aplicación como un filtrado por ubicación física, puesto o un cambio en el formato de las bitácoras de envío.

¹³ Son campos de una tabla a través de los cuales se puede tener acceso a otra. Más adelante, se verá este aspecto con mayor detalle.

¹⁴ Como los proporcionados por compañías como Yahoo, Google o Microsoft.

2.2 Metodologías de desarrollo.

Metodología es el "conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevas aplicaciones informáticas"¹⁵.

Desde la concepción de la programación estructurada por Dijkstra en 1968 surgieron técnicas como la "Warnier Jackson" y metodologías como la "Myers Yourdon". Luego de un largo proceso evolutivo que incluyó el paso a sistemas en tiempo real y Métrica, se ha llegado a las metodologías orientadas a objetos y algunas más revolucionarias como la programación extrema.

Algunas de éstas metodologías como la SSADM¹⁶ reflejan la necesidad de interactuar de manera estrecha con el cliente además de establecer una secuencia de procesos bien definida. Además, propone la división del sistema en datos, eventos y procesos. Sin embargo, SSADM proporciona un conjunto de procedimientos de análisis y diseño, pero no cubre aspectos como planificación ni construcción de código.

2.2.1 Metodología Yourdon

Modelo de ambiente: Se determina qué pertenece al interior y qué al exterior del sistema. También se definen las interfaces entre el sistema y el usuario. Básicamente se toma en cuenta cuál información entra y cuál sale del sistema. Para desarrollar este modelo es necesario:

- a) Declaración de propósitos. Va dirigida a personas de nivel administrativo. Consta solo de algunos renglones que establezcan los límites del sistema, su interacción con otros sistemas y los beneficios esperados
- b) Lista de eventos. Contiene los acontecimientos del mundo exterior a los cuales debe dar respuesta el sistema. Indican el tipo de evento: F de flujo de datos, C de control (momento impredecible) y T temporal. Puede incluir eventos de fallo.
- c) Diagrama de contexto DFD con 1 burbuja. Enfatiza las personas, organizaciones y sistemas con los que se comunica el sistema (terminadores) representados con rectángulos. Los datos de entrada y los de salida.

¹⁵ Curso Ingeniería del Software <http://www.eduangi.com/ingsw/tema2/2-4.htm>

¹⁶ Desarrollada por la Agencia Central de Telecomunicaciones y Cómputo del gobierno británico.

Modelo de comportamiento:

Sirve para identificar respuestas a eventos. El proceso consiste en:

- 1) Dibujar una burbuja para cada evento de la lista y nombrarla describiendo la respuesta del sistema al evento
- 2) Dibujar las entradas y salidas para que la burbuja dé la respuesta requerida. Los almacenes deben ser tomados en cuenta.
- 3) El diagrama de flujo obtenido se compara con el de contexto y la lista de eventos.

A partir del diagrama obtenido se agrupan procesos relacionados y se representan con una sola burbuja, en un nivel superior tomando en cuenta que los procesos manejan datos relacionados. Si existe un conjunto de procesos que hacen referencia a un almacén y no hay otros procesos que lo usen, se crea una burbuja de nivel superior para agrupar los procesos y esconder el almacén. Si un proceso tiene muchos flujos de entrada/salida, se debe intentar la creación de grupos de flujos, lo que dará origen a una burbuja de nivel inferior.

- a) Diagrama E-R: Define entidades y atributos en el Diccionario de Datos
- b) Diccionario de datos completo: Incluye descripciones de procesos primitivos
- c) Se debe analizar la relación costo – beneficio y los tiempos de desarrollo

Además del sistema concluido, debe ser entregada cierta documentación como el análisis de hardware y software ya sea que requiera ser adquirido o se cuente con él. La especificación de procesos, diagrama entidad relación, manual de usuario y técnico y el diccionario de datos entre otros.

2.2.2 Proceso Unificado

Es un proceso de desarrollo de software configurable y adaptable a proyectos de distinto tamaño y complejidad. Está basado en tecnología orientada a objetos para el desarrollo de programas de misión crítica¹⁷.

Guía a los desarrolladores en la administración del desarrollo iterativo controlado mientras

¹⁷ Sistemas de los que una compañía u organización depende funcional y/o económicamente.

balancea los requerimientos del negocio, tiempo al mercado y riesgos del proyecto. Describe los pasos involucrados en la captura de requerimientos y el establecimiento de una guía arquitectónica rápida para el diseño y prueba del sistema.

El proceso unificado tiene un enfoque caracterizado por:

- ◆ Interacción continua con el usuario
- ◆ Previsión de riesgos
- ◆ Liberaciones frecuentes
- ◆ Aseguramiento de la calidad
- ◆ Involucrar a todo el equipo de desarrollo en la toma de decisiones
- ◆ Anticipación en el cambio de requerimientos

El proceso unificado "define quién está haciendo qué, cuándo lo hace y cómo alcanza cierto objetivo"¹⁸. Entre las características con que debe cumplir están el ser actual y actualizable. Se basa en tres ideas básicas: está dirigido por casos de uso¹⁹, los cuales deben elaborarse en conjunto con el usuario y desde la perspectiva de éste; está centrado en la arquitectura del sistema y es iterativa²⁰ e incremental. Su ciclo de vida está definido en cuatro fases:

1. Inicio
2. Elaboración
3. Construcción
4. Transición

Cada una desarrolla cinco flujos de trabajo: requerimientos, análisis, diseño, implementación y prueba.

Los principales motivos por los que se ha hecho tan popular es que es intuitivo y sistemático en la captura de requerimientos funcionales desde el punto de vista de lo esperado por el usuario y el proceso es dirigido al análisis, diseño y pruebas.

18 Jacobson. Applying UML in The Unified Process.

19 La especificación del comportamiento de un sistema o parte de él para cubrir una función requerida por un usuario.

20 Breve descripción del sistema, identificación de clientes, definición de objetivos, funciones y atributos del sistema, suposiciones, riesgos y dependencias.

Los casos de uso:

- ◆ Proporcionan una forma intuitiva de comunicación con el usuario, lo que propicia el trabajo en conjunto usuario - cliente – desarrollador.
- ◆ Limitan al sistema.
- ◆ Ayudan a encontrar las clases y a definir la interfaz de usuario del sistema.
- ◆ Ayudan a la creación del Manual de Usuario pues definen de qué manera los usuarios van a interactuar con el sistema.

El proceso unificado, también toma en cuenta la arquitectura del sistema, que especifica qué elementos “clave” describen al sistema y sus características (interfaces, comportamiento y colaboraciones) y la forma en que todo ello se organiza, por ejemplo bajo el patrón Cliente/Servidor, Peer-to-Peer o una combinación de ellos. Debe considerar partes del sistema que pueden ser reutilizadas de sistemas heredados, además de considerar requerimientos no funcionales que restrinjan al sistema en recursos, políticas, estándares y lo orienten a cumplir determinadas características.

La arquitectura se crea a partir de la segunda fase que marca el Proceso Unificado, y se forma a partir de vistas de los modelos del sistema.

2.2.3 Extreme Programming

Se trata de una metodología de reciente creación que nació en Chrysler y sigue la filosofía de las metodologías ágiles que buscan simplificar al máximo las tareas de ingeniería, tienen muy pocas reglas y técnicas²¹. Su objetivo es obtener la máxima satisfacción del usuario y una respuesta rápida a cambios en los requerimientos. Se recomienda para proyectos donde los requisitos pueden cambiar de manera inesperada.

No proporciona ninguna herramienta para comprender la situación del usuario y su entorno. La colaboración del cliente es fundamental.

Las máximas en la programación extrema y las metodologías extremas son:

- ✓ La prioridad es la satisfacción del cliente mediante entregas continuas.
- ✓ Los cambios son bienvenidos y asimilados de manera inmediata.
- ✓ Los intervalos entre las entregas de software son menores a un mes.

²¹ A diferencia de las clásicas que tienen muchas reglas, documentos y pasos a seguir, además de requerir mucha disciplina y tiempo. Utilizan herramientas CASE y su documentación en ocasiones resulta abstracta y complicada.

- ✓ Los usuarios y desarrolladores trabajan juntos durante todo el proyecto.
- ✓ El equipo de desarrollo debe contar con apoyo, confianza y un buen entorno de trabajo.
- ✓ Para la comunicación de información, el mejor medio es hablarlo directamente.
- ✓ La medida del progreso del proyecto es el software que funciona.
- ✓ La simplicidad es esencial.
- ✓ Las mejores arquitecturas y diseños, surgen de equipos organizados por sí mismos.
- ✓ El equipo ajusta su comportamiento en base a una autoreflexión de la situación.

La programación extrema como parte de las metodologías ágiles está basada en entornos dinámicos, grupos menores a 10 personas, comunicación continua y directa y lo más importante: está orientado a la codificación.

Si bien, no existe una delimitación de funciones rígida, existen algunos roles que pueden ser determinados:

Programador	Es el responsable de las decisiones técnicas y la construcción del sistema. No existe distinción entre analistas, diseñadores y codificadores. El programador es quien programa, diseña y prueba.
Jefe del proyecto	Organiza las reuniones y asegura condiciones adecuadas.
Usuario	No es ajeno al equipo de trabajo. Determina qué construir y cuándo y establece las pruebas funcionales.

Tabla 6 Roles en la programación extrema.

Además de estos tres personajes existen algunos otros como el encargado de pruebas que asiste al usuario en las pruebas funcionales o el rastreador que es quien lleva una bitácora de los avances.

Propone un desarrollo iterativo compuesto por 4 pasos en los que se añaden nuevas funcionalidades a la aplicación. En ningún momento establece pasos o técnicas estrictas, solo propone un conjunto de recomendaciones y consejos que facilitan el desarrollo. Los cuatro pasos propuestos son:

1. Planificación

En esta etapa, el cliente y los desarrolladores "negocian" las historias²², se trata de implementar primero las historias más importantes bajo el criterio del cliente. Es muy importante solo realizar los desarrollos planeados.

Deben haber reuniones diarias para la solución de dudas o problemas y todos los integrantes se involucran en todas las partes del proyecto.

2. Diseño

Es clave que el diseño sea simple. Se realiza la elección de una "metáfora del sistema" para evitar diferencias entre nombres de clases y métodos. El diseño se realiza mediante tarjetas CRC (Clases, responsabilidades y colaboraciones)²³.

El diseñador construye sistemas tan simples como sea posible para dar solución a los problemas más importantes. De hecho, estos prototipos sirven en muchas ocasiones como base al sistema final.

La reutilización del código es recomendada siempre y cuando, exista un análisis previo del mismo. Es vital la eliminación de código redundante o sin uso. De nuevo se trata de mantener código tan simple como sea posible.

3. Codificación

El código se implementa de acuerdo a los estándares de programación del lenguaje utilizado. Una tarea primordial es siempre la optimización del código. La primera tarea es generar las unidades de prueba.

Todo el código debe ser creado por una pareja de programadores en una estación de trabajo única. Mientras uno se encarga en la creación de cada método, otro estudia la forma de incrustarlo dentro de la clase.

La integración del código con el resto del sistema no debe ser tarea de una persona específica, puede implementarse la técnica de código compartido²⁴ e integración secuencial de código²⁵. Es importante que los programadores además de insertar su código, incluyan *unidades de prueba* que cada clase debe superar para asegurar que funciona correctamente.

22 Establecen los requisitos, se asignan tareas de programación con el tiempo de desarrollo en horas.

23 Cada tarjeta CRC representa un objeto y en la sesión para su diseño, se simulan los mensajes que se envían unos a otros. Los diseñadores crean nuevas tarjetas, las modifican y sugieren mejoras.

24 Cualquier programador puede añadir alguna funcionalidad o corregir errores a cualquier parte del proyecto.

25 Solo una pareja integra su código a la vez y lo realiza sin importar que el avance sea mínimo; esto garantiza que todos estén actualizados.

4. Prueba

Luego vienen las pruebas de aceptación creadas a partir de las historias del usuario. Se trata de cajas negras con una entrada que conlleva a una salida esperada.

¿Documentación? Uno de los "lineamientos" que se deben seguir es no realizar documentación escrita. El único documento es el código fuente. De nuevo, es necesario establecer una comunicación directa entre usuario y equipo desarrollador.

2.3 Sistemas orientados a web.

Desde hace algunos años, en el ámbito del desarrollo de aplicaciones, se ha dado un giro total en la manera de programar. Herramientas como Informix o Sybase eran utilizadas por gran cantidad de empresas como base en el desarrollo de sus sistemas dedicados.

Ese tipo de aplicaciones, era "orientada a conexión", lo que implica que la conexión al servidor de la base de datos era persistente ya sea que se estuviera explotando o no. Esto era causante de serios problemas, debido a la existencia de múltiples hilos de conexión (muchos de ellos ociosos) que podían saturar no sólo a la misma aplicación (incluida la base de datos) sino al servidor mismo e incluso a toda la red involucrada.

Otro de los aspectos que se deben considerar, es la necesidad en éste tipo de programación de hacer el diseño y desarrollo de la misma ventana de la aplicación (lo que como veremos más adelante, ya no es forzoso).

En el desarrollo de aplicaciones orientados a la web, se cuenta ya con lenguajes de desarrollo (en su gran mayoría interpretados) como Perl, Java²⁶, ASP o PHP que ofrecen la posibilidad de desarrollo tanto estructurado como orientado a objetos.

Las ventajas que ofrece una aplicación "orientada a no conexión" como son las implementadas a través de web son entre otras:

- ✓ El hilo de conexión se crea y destruye bajo demanda, es decir, cuando un cliente realiza una petición, se abre el hilo y cuando ésta es respondida, se cierra de nuevo dando paso a una nueva petición proveniente del mismo u otro cliente.
- ✓ En una aplicación desarrollada para la web, no es necesario preocuparse por la programación del entorno, ya que el navegador se encarga de la interpretación de etiquetas que dan forma a la presentación.
- ✓ El sistema es multiplataforma al no depender de algún sistema operativo en el cliente o de la instalación de paquetería adicional (incluida la misma aplicación).

Para generar o escribir el código de las páginas, es necesario utilizar algún editor ya sea de propósito general o específicamente para el diseño de páginas web. Por ello, en el siguiente apartado veremos un análisis de algunos de los editores disponibles para ello para después entrar de lleno al análisis de plataformas.

²⁶ Java es un lenguaje tanto interpretado como compilado. Un programa en Java se compila bajo un formato binario independiente de plataforma y puede también ser interpretado por un ambiente de runtime de Java específico para la plataforma.

2.4 Editores de HTML

Existen dos maneras de realizar una página web. La primera es escribir el código html en editores de texto comunes como vi y pico (Unix/Linux) o el editor de notas (Windows). Otra es utilizar editores gráficos que en ocasiones generan el código de manera automática. Las principales diferencias entre usar un editor de texto común y una herramienta gráfica son:

Editor de texto	Editor gráfico
Existe un mejor dominio del código. Se evita la generación de código redundante y basura en la página.	El código es de mala calidad, pues incluye etiquetas innecesarias y normalmente no realiza un correcto sangrado de los bloques.
Es más complicado, lento y difícil, lo que conlleva entre otras cosas al uso de más tiempo.	Es sencillo de aprender.
El aprendizaje de código HTML es más profundo, con lo que las limitantes son menores.	La velocidad en la generación de la página es mucho mayor.

Tabla 7 Diferencias entre editores gráficos y de texto

Normalmente, no se puede excluir a alguna de las dos vertientes. Si se desarrolla en editores de texto, tarde o temprano se necesita de una mayor velocidad y es necesario recurrir a un editor gráfico, por el contrario, si se desarrolla de manera gráfica, hay detalles prácticamente imposibles de realizar sin ir directamente al código.

Existen herramientas que no pueden ser clasificadas en ninguno de los dos tipos de editores. El código normalmente debe ser escrito por el desarrollador, pero al ser editores especialmente diseñados para la creación de páginas web, comúnmente completan de manera automática las etiquetas que se están escribiendo o pueden ser insertadas desde un menú. Despliega las etiquetas en distintos colores haciendo más sencilla la localización de instrucciones; además, cuentan con visualizadores de página, con lo que no es necesario recurrir al navegador.

Los editores gráficos son conocidos como WYSIWYG²⁷, pues la edición se realiza simplemente insertando el texto con el formato deseado, las imágenes, etc. y el programa se encarga de generar el código html.

Entre los editores más conocidos dentro de estas categorías se encuentran:

²⁷ What You See Is What You Get (lo que ves es lo que consigues).

Editores de texto	
Note pad	Editor de texto para la creación de documentos sencillos de Microsoft. Su extensión por defecto es .txt. Permite el uso de distintos juegos de caracteres.
vi	Editor de texto para la generación de todo tipo de texto plano, especialmente útil en la edición de programas. Requiere del constante cambio entre modo inserción y edición. Los sistemas operativos UNIX y Linux cuentan con él.
pico	También disponible en UNIX, está basado en el entorno de edición de pine (un cliente de correo en texto plano muy común en Unix)
Editores híbridos	
Home Site	Ligero editor de código para desarrollo web. Soporta los formatos HTML, CFML, JSP, y XHTML. Es desarrollado por la compañía Macromedia
UltraEdit	Editor con versiones de 16, 32 y 64 bits para Windows desarrollado por IDM.
Quanta Plus	Herramienta de desarrollo web para el ambiente KDE. Intenta mantener un balance entre los más populares lenguajes de marca y CSS.
Editores WYSIWYG	
Dreamweaver	Es una combinación de herramientas visuales de disposición, características de desarrollo de aplicaciones y soporte para la edición de código. Proporciona integración y diseño basado en CSS. Es realizado por Macromedia.
Front Page	Incluye las herramientas de diseño, creación y publicación para la creación de sitios dinámicos. Señala contar con el primer editor comercial totalmente WYSIWYG para transformaciones de lenguaje extensible de hojas de estilo (XSLT). Por Microsoft.
Mozilla Composer	Al igual que otros navegadores, Mozilla cuenta con su propio entorno de desarrollo con características WYSIWYG.

Tabla 8 Editores de texto, híbridos y wysiwyg.

La etapa de diseño gráfico²⁸ se realizará en *Dreamweaver*. El proceso restante será realizado con *vi* debido a la facilidad de edición de archivos de manera remota (directo en el servidor) y al conocimiento previo de sus características y comandos.

²⁸ Realizada por el personal del Área de Desarrollo Web.

2.5 Plataformas para desarrollo web

En el nivel más bajo de la arquitectura cliente-servidor, el proceso servidor crea un socket²⁹ a través del cual el cliente accesa a una aplicación. El servicio es pedido a través de un número de puerto³⁰ que es responsable principalmente de la comunicación entre cliente y servidor.

La comunicación web comienza cuando un cliente (digamos un navegador) hace una petición a un servidor web enviando una URL³¹ incluyendo el tipo de servicio que solicita (http). El servidor toma la solicitud y si cuenta con el servicio y recurso solicitados, envía una respuesta al cliente que se encarga de interpretar la información recibida.

Existe una gran cantidad de servidores de web entre los que se puede contar a IIS y Apache.

Para éste sistema, es necesario el uso de una base de datos. Un RDBMS (Sistema manejador de bases de datos relacionales), se ha vuelto indispensable en muchas empresas tanto públicas como privadas para la gestión de información. Entre los sistemas de bases de datos actuales algunos de los mas conocidos son SQL Server de Microsoft, PostgreSQL, Oracle y MySQL.

Las plataformas de mayor uso en la actualidad son:

<i>Plataforma</i>	<i>Tecnología</i>
.NET	La plataforma de Microsoft soporta tecnología ASP.
Java	Sun permite el uso de servidores web independientes como Apache en combinación con lenguaje Java.
SL	La opción del software libre, permite el uso de lenguajes como Java, PHP, Perl, Python e incluso ASP

Tabla 9 Plataformas web.

La tecnología dentro del servidor es un factor importante a tomar en consideración, sin embargo, otra parte importante a tomar en cuenta, debe ser el manejador de bases de datos (DBA). Existe una gran cantidad de opciones en este ramo, algunas son muy costosas como Oracle o Sybase, otras pueden ser consideradas como de mediano coste como SQL Server de Microsoft y existe también

29 Punto de comunicación lógico (en el caso cliente-servidor). Es generado a través de llamadas al sistema. Para el acceso se requiere: nombre del servicio (http, telnet), protocolo (TCP) y número de puerto (80, 23).

30 80 para http.

31 Uniform Resource Locator (localizador uniforme de recurso). Es una cadena de caracteres con la que se asigna una dirección única a cada recurso de información (página) de internet.

la opción de manejadores con versiones libres³² como PostgreSQL y MySQL.

Existen ciertas relaciones entre plataformas y DBA. Por ejemplo, .net con SQL Server o MySQL con Apache³³

Una de las opciones con mayor fuerza en el ámbito del software libre es la tecnología denominada LAMP (Linux - Apache - MySQL - PHP) y para programación en JSP, es común el uso de Apache en conjunto con Tomcat³⁴ e indistintamente PostgreSQL o MySQL.

Luego del impacto de los lenguajes compilados (C, Pascal, etc.) se ha dado una nueva dirección en cuanto a desarrollo se refiere gracias a la aparición de Internet. De hecho, se han creado lenguajes estructurados y orientados a objetos específicamente para el desarrollo de aplicaciones web (en su mayoría interpretados y embebidos en código html).

A continuación una breve descripción de las características de cada una de las plataformas.

2.5.1 .net

Permite el uso de lenguajes como Visual Basic, lo que permite una alta velocidad de desarrollo y para personas habituadas al lenguaje, un sencillo aprendizaje.

Su administrador de bases de datos de facto es SQL Server que promete ser potente, flexible y fiable.

Esta plataforma en su totalidad cuenta con el respaldo de la que es tal vez la empresa más poderosa y de mayor cobertura en el ámbito del software: Microsoft, lo que conlleva un respaldo que puede ser catalogado como bastante serio. Sin embargo, dentro de este análisis, encuentro tres grandes aspectos en contra:

- x La adquisición de la plataforma completa resulta costosa.
- x Muchos de los virus informáticos, están orientados a esta plataforma.
- x Continuamente, se descubren nuevas vulnerabilidades.

32 Bajo la GNU General Public License de la FSF, orientada a términos de distribución, modificación y uso de software.

33 El servidor web de mayor uso a nivel mundial.

34 Es un servidor aplicativo que tiene un Runtime de Java, un Web Container y un motor de servlets que es utilizado para permitir al servidor web "servir" aplicaciones de Java ya que éste no cuenta por sí mismo con un módulo que realice esta tarea.

2.5.1.1 Servidor IIS

Es un componente opcional de Windows en su versión de servidor, actualmente se encuentra en la versión 6.0 (incluida en Win Server 2003).

Está diseñado para trabajar bajo un ambiente Windows -lo que resulta una limitante- aunque también conlleva algunos beneficios como una operación bastante integrada y en comunión con el sistema operativo además de permitir una administración más sencilla a través de herramientas del sistema operativo anfitrión.

Cuenta con un ambiente dinámico de desarrollo (lo que permite la incrustación de código en páginas HTML). El sistema ASP permite trabajar en lenguajes como Visual Basic o JavaScript. Tiene soporte para CGI's³⁵.

Debido a la gran integración con el sistema operativo, la misma configuración de usuarios y grupos de este, es utilizada dentro del servidor en aspectos de seguridad y autenticación. Esto permite al sistema dar acceso a usuarios de una intranet suponiendo que ya están autenticados dentro del ambiente Windows.

Por último, se debe señalar que el servidor IIS en conjunto con el sistema operativo Windows, es costoso, y no hay manera de obtenerlo de manera gratuita.

2.5.1.2 SQL Server

SQL Server es un paquete que incluye al manejador de bases de datos y un analizador de datos. Está perfectamente habilitado para ambientes web y es compatible con XML. Cuenta con un alto nivel de escalabilidad³⁶ y confiabilidad.

Es capaz de optimizar y depurar consultas en forma interactiva, además de la manipulación de datos y la codificación visual de aplicaciones a través de Visual Studio. Al igual que IIS, está totalmente integrado al sistema operativo.

2.5.1.3 ASP

La tecnología de Microsoft ASP (Active Server Pages) permite la creación de páginas dinámicas con la ejecución de scripts del lado del servidor. Estos scripts pueden ser escritos en VBScript y JavaScript.

³⁵ Common Gateway Interface (Pasarela de Interfaz Común). El servidor web pasa las solicitudes del cliente a un programa externo, la salida de dicho programa es enviada al cliente en lugar del archivo estático tradicional.

³⁶ Aprovecha los sistemas de multiproceso simétrico (SMP, *Symmetrical Multiprocessor*). Y puede utilizar hasta 32 procesadores y 64 GB de RAM.

Al hablar de ASP, nos referimos a una tecnología propietaria que implica su uso forzosamente en combinación con productos de Microsoft como IIS y Windows en el servidor. Algo de lo más reciente, es la tecnología denominada ASP.net que no es totalmente interpretada, sino que incluye programas compilados que se benefician de las ventajas de éstos sobre los interpretados.

La mayor ventaja de esta plataforma es la excelente integración entre los servidores de página web, base de datos y el intérprete.

2.5.2 LAMP

2.5.2.1 Linux

Es el sistema operativo con mayor crecimiento en todo el mundo durante la última década. Lo peculiar cuando se habla de éste SO es que a diferencia de otros como Windows o Solaris, no es desarrollado por una compañía en particular, lo que conlleva a la existencia de distintas distribuciones³⁷ con objetivos y características muy diversas. Linus Torvalds y su equipo se encargan de desarrollar el kernel para luego liberarlo y que las diversas organizaciones implementen sobre él las aplicaciones correspondientes.

Como se trata de un SO que en su inicio estuvo basado en UNIX y (Minix), mantiene en la actualidad muchas similitudes con esta familia de sistemas operativos³⁸ desde los comandos básicos, los de administración o los *file systems* en los que está organizado el sistema en su totalidad. Incluso hay quien señala que Linux es el "heredero" directo de UNIX y que la tendencia es que los usuarios de UNIX paulatinamente lo abandonen y adopten a Linux³⁹. Como UNIX-like, Linux es robusto, multiusuario, multitarea y con una administración apropiada puede ser bastante seguro⁴⁰.

El principal enemigo de Linux es la costumbre en el uso de otro sistema operativo (Windows) y que debido a la gran cantidad de distribuciones existentes y en ocasiones a la falta de estandarización, sobre todo los puestos de mando de las empresas no confían aún lo suficiente en él. Sin embargo, diversas organizaciones desarrolladoras de distribuciones Linux, han tratado de establecer empresas que signifiquen al usuario un respaldo y la garantía necesaria para la adopción del sistema⁴¹.

37 Se considera que existen más de 100, algunas de las más conocidas son Red Hat, Mandrake, SuSE y Debian.

38 Solaris de SUN, AIX de IBM, IRIX de SGI, MacOSX de Macintosh y HP-UX entre otros.

39 Desde mi perspectiva personal, esto no es cierto (al menos en la actualidad).

40 No es mi afán extenderme demasiado en la descripción de estos aspectos. Existe mucha bibliografía al respecto como *Unix* de Steve Moritsugu o *Unix a fondo* de James Armstrong.

41 Solo como un ejemplo, podemos mencionar a Novell que se ha unido con SuSE, una de las distribuciones más fuertes en la actualidad y Ximian, la empresa responsable de uno de los escritorios más usados en Linux (Gnome) y diversas aplicaciones más.

2.5.2.2 Servidor Apache

Por su parte, Apache (cuya versión más reciente es la 2.0.50) es un producto nacido bajo UNIX que luego fue diversificando su campo de sistemas operativos que en la actualidad incluye al propio Windows y MacOSX. Esto ocurre gracias al APR⁴² que permite al núcleo de Apache correr prácticamente en cualquier sistema operativo siempre y cuando cuente con compilador de lenguaje C.

Es seguro y estable además de ser bastante eficiente en el manejo de peticiones. Puede ser personalizado de acuerdo a las necesidades de cada caso, desde distintas configuraciones para cada página del servidor hasta soporte de lenguajes interpretados.

Cuenta además con módulos de multiproceso que reciben y procesan las peticiones. Bajo en modelo de CGI, Apache es capaz de trabajar con una buena cantidad de lenguajes. Sin embargo, la mejor opción es el uso de módulos dinámicos que incorporan el interprete directamente a Apache lo que implica un beneficio en la velocidad de respuesta en lenguajes como PHP, Python o Perl.

La administración y seguridad de Apache es totalmente independiente del sistema operativo. Sin embargo, existen diversos módulos y adaptadores capaces de soportar varios recursos de autenticación y seguridad como el propio archivo *passwd* de un sistema Unix o el uso de un servidor LDAP. Una característica importante de Apache es que es libre, lo que conlleva a contar con el código fuente y una enorme cantidad de desarrolladores detrás.

2.5.2.3 MySQL

Es un manejador de bases de datos cliente-servidor que "incluye un servidor SQL, programas cliente para acceder al servidor, herramientas administrativas y una interfaz de programación para la escritura de programas"⁴³. Entre sus clientes se encuentran Yahoo, Google, Cisco y el Laboratorio de Propulsión de la NASA.

Es un DBA gratuito (a menos que se busque comercializar en conjunto con alguna aplicación desarrollada). Es posible ejecutarlo en sistemas operativos propietarios (como Windows o Solaris⁴⁴) y libres como Linux. Tiene capacidad para el manejo de datos con millones de registros. Un buen grupo de personas (la mayoría de ellos desarrolladores de aplicaciones) sostiene que se trata del manejador de bases de datos más rápido que existe, lo que se atribuye a su falta de garantía en la integridad referencial⁴⁵, lo que para aplicaciones críticas y grandes, puede ser una desventaja con respecto a competidores como PostgreSQL (también catalogado dentro del software libre) u Oracle.

42 Apache Portable Runtime.

43 DuBois P. MySQL, edición especial, p.XXIII

44 De hecho, la primera versión de MySQL fue desarrollada para ese sistema operativo

45 A grandes rasgos, cuando las tablas están relacionadas, la eliminación o modificación en alguna de las tablas, puede afectar a otra, MySQL no se asegura de que los cambios sean reflejados en las demás tablas o no las afecten.

Una de las más importantes características de MySQL es el soporte del mismo SQL⁴⁶ (Structured Query Language) y la interconectividad con aplicaciones que utilizan el protocolo ODBC⁴⁷ (Open Database Connectivity). Es capaz de atender conexiones simultáneas además de presentar diversas bases de datos también de manera simultánea.

Además del cliente para MySQL, cuenta con interfaces para acceso desde lenguajes como C, Java o PHP. En total, el servidor provee de:

- El motor del servidor SQL
- Programas cliente para acceso al servidor. Es posible introducir consultas directamente desde el programa además de contar con utilerías de administración; incluso cuenta con buenas herramientas de importación y exportación de datos.
- Biblioteca cliente para la escritura de programas en C y enlaces para otros lenguajes.

Es un administrador de bases de datos relacional con excelente comportamiento en aplicaciones orientadas a web. Incluso, existen herramientas gráficas⁴⁸ que permiten su administración sin el uso de línea de comandos.

Si se desea, incluso existen planes de licenciamiento y diversas listas de desarrolladores.

2.5.2.4 PHP

Es un acrónimo de "PHP: Hypertext Preprocessor" y es un proyecto de la Apache Software Foundation⁴⁹. Consiste en un lenguaje de código abierto interpretado de alto nivel (parecido a C y Pascal) embebido en HTML y ejecutado del lado del servidor. Es capaz de realizar cualquier cosa que se pueda hacer con un CGI como la creación de páginas dinámicas o el manejo de formularios.

PHP participa en tres campos principales:

- Scripts del lado del servidor. El uso mas común de PHP. Necesita un intérprete de PHP, un servidor web y un navegador del lado del cliente.
- Scripts en línea de comandos. Un script de PHP se puede correr desde línea de comandos sin necesidad del servidor web. Estos scripts, se usan por lo regular en archivos ejecutados por el *cron* del sistema operativo o en procesamiento de texto.
- Aplicaciones de interfaz gráfica. Tal vez no es su mejor aplicación, pero es posible utilizar la combinación PHP-GTK para escribir aplicaciones gráficas.

46 Lenguaje de consulta estructurado.

47 Protocolo de comunicación desarrollado por Microsoft.

48 Una de las más conocidas es phpMyAdmin.

49 La misma que produce al servidor Apache y varios productos más.

Puede utilizarse bajo sistemas operativos tan variados como Linux, Unix, Windows y MacOSX. Soporta la mayoría de servidores web incluyendo Apache, IIS e iPlanet de Netscape. También tiene la posibilidad de usar programación estructurada u orientada a objetos.

Puede interactuar con muchos manejadores de bases de datos como Oracle, MySQL, dBase, Informix, PostgreSQL, etc. además de soportar ODBC.

Una de las ventajas sobre sus competidores (ASP, Perl) es su orientación a objetos incrustada en código HTML, que es muy rápida y su portabilidad es bastante buena. Puede correr como un módulo del propio Apache y por si fuera poco, es sencillo de aprender para programadores con alguna experiencia en lenguajes como C o Pascal.

PHP también es capaz de crear imágenes, archivos PDF y archivos Flash por medio de bibliotecas. Incluso, puede comunicarse con protocolos como IMAP, POP3, HTTP, y varios más.

Existe una variante de la plataforma LAMP denominada arquitectura Java que incluye JSP, servlets, aplicaciones Java y J2EE bajo sistema operativo Linux, servidor web Apache y Tomcat o Jboss como ligadores entre Apache y Java. Comúnmente el manejador de bases de datos utilizado en esta plataforma es PostgreSQL. Posiblemente el problema en esta plataforma sea la complejidad del lenguaje Java, lo que conlleva a una curva de aprendizaje y desarrollo más prolongada.

2.5.3 Algunas opciones independientes de plataforma.

2.5.3.1 Oracle

Es una herramienta de administración de bases de datos cliente-servidor. Está más bien dirigido a grandes empresas debido a su alto precio y su potencia. El lenguaje utilizado para la interacción con bases de datos Oracle es PL/SQL o SQL para aplicaciones de formulario.

En la actualidad se encuentra en la versión 10 y cuenta con dos herramientas básicas de desarrollo:

- ♦ Oracle Developer. Permite crear formularios y compilarlos de manera local para luego compartir los archivos resultantes. Su principal ventaja es ser intuitiva y que la creación de formularios es muy parecida a la de lenguajes visuales.
- ♦ Oracle Designer. Es una herramienta que se conecta a la base de datos para crear los formularios en ella. Por medio de él, los usuarios se conectan a la aplicación y no es necesario la compartición "manual" de cada programa. Sin embargo, no cuenta con un entorno gráfico de desarrollo del formulario.

2.5.3.2 JAVA

Es un lenguaje de programación de SUN Microsystems que se desarrolló buscando competir con Microsoft y se ha convertido en estándar para el desarrollo de aplicaciones basadas en Internet. Sus versiones preliminares, estaban derivadas de C++.

Cuenta con una buena combinación entre construcción de objetos y diseño de sistemas gráficos. Estos componentes están orientados al desarrollo de sistemas de información basados en redes.

Java es un lenguaje sencillo (aunque no debemos olvidar que un lenguaje orientado a objetos es algo intrínsecamente complejo). Se trata de un lenguaje independiente de plataforma y bastante seguro en aspectos de acceso a memoria. Tiene características multihilo, es robusto y está bastante bien integrado al protocolo TCP/IP.

La mayor peculiaridad de Java es que es considerado un lenguaje medio interpretado, medio compilado debido a que el código fuente se compila a un lenguaje intermedio que si bien es cercano al lenguaje máquina. es independiente del hardware y el sistema operativo. A esto se le llama bytecodes. Luego, el lenguaje intermedio se interpreta por la máquina virtual de Java.

A su aparición, era considerado varias veces más lento que C++ (alrededor de 20), en la actualidad la diferencia se ha visto reducida y se habla de una lentitud solo de 2 a 3 veces mayor a la de C++.

Al ser un lenguaje completamente orientado a objetos, cumple con cinco características básicas:

- ♦ Todo es un objeto. Cada elemento es modelado como un objeto.
- ♦ Un programa es un conjunto de objetos. A un objeto se le envía un mensaje para que realice alguna acción. Es decir, se ejecuta un método del objeto.
- ♦ Cada objeto tiene su propia memoria. Puede contener otros objetos, lo que incrementa la complejidad.
- ♦ Todo objeto tiene un tipo. Cada objeto es una instancia (caso particular) de una clase (tipo general)
- ♦ Todos los objetos de un tipo pueden recibir los mismos mensajes.

Solo basta señalar que algunos de los elementos de las plataformas tratadas, pueden tomarse de manera independiente, por ejemplo, se podría levantar un servidor Apache con MySQL y PHP sobre un sistema operativo Windows (lo que no se puede hacer con IIS). Java gracias a su máquina virtual, puede comunicarse con cualquier plataforma.

La Programación Extrema puede ser implementada sobre cualquiera de las plataformas tratadas. Gracias a su metodología de desarrollo, en cualquier caso cada integrante del equipo puede tener acceso al espacio de desarrollo (servidor) simplemente con la cuenta apropiada. Basados en la inclusión de archivos (bibliotecas) en cada uno de los archivos del programa, la interacción entre los desarrolladores y sus módulos resulta sumamente sencilla con la comunicación apropiada.

2.5.4 Decisión de la plataforma.

Estas tres plataformas cuentan con muchas otras características tanto favorables como desfavorables que serán resumidas en los siguientes cuadros comparativos⁵⁰.

	<i>.net</i>	<i>LAMP</i>	<i>Java</i>
Dependencia con el servidor	dependiente	pseudo-dependiente	independiente
Proceso de desarrollo	interno	open source	open source
Etiquetas	limitadas	limitadas	extensibles
Reusabilidad entre plataformas.	no	sí	sí
Tipo de lenguaje	interpretado	interpretado	compilado

JSP se puede ejecutar en distintos sistemas operativos y servidores web. .net solo es soportado por servidores IIS y Personal Web Server de Microsoft⁵¹. LAMP puede ser "seccionado" y convertirse en AMP con lo que se implementa bajo UNIX e incluso Windows

El API⁵² JSP y los cuatro componentes de LAMP son desarrollados por comunidades mundiales mientras que la tecnología .net es desarrollada al interior de Microsoft.

JSP además de usar etiquetas preestablecidas, permite la creación de nuevas etiquetas.

Los componentes JSP y PHP pueden ser portados a otros sistemas operativos o servidores de web. Basta con implementar el servidor apropiado para interpretarlos.

Java es compilado en Servlets⁵³ que permiten el acceso a bibliotecas y servicios. ASP y PHP son interpretados durante la ejecución

Tabla 10 Comparación: .net, LAMP, Java.

50 Basados en análisis contenidos en: Brown, Martin. IIS vs. Apache, Looking Beyond the Rhetoric. Chamorro Villar, Ricardo Análisis comparativo entre Microsoft .NET y SUN J2EE. Desarrollo web jsp vs asp.

51 Existen proyectos como Ximian MONO que ofrecen las características de .net con algunas adicionales.

52 Interface de Programación de Aplicaciones. Es un conjunto de especificaciones de comunicación entre componentes software. Uno de los principales propósitos de una API es proporcionar un conjunto de funciones de uso general, por ejemplo evitando a los programadores el desarrollo de aplicaciones desde cero.

53 Objetos que corren dentro del contexto de un servidor de aplicaciones como Tomcat y se ejecutan en un servidor web. El uso más común de los *servlets* es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

	<i>PHP</i>	<i>ASP</i>	<i>JSP</i>
Costo del software	gratuito	gratuito	gratuito
Costo de la plataforma	gratuito	No es gratuito	gratuito
velocidad	alta	baja	alta
eficiencia	alta	baja	alta
seguridad	alta	alta	alta
plataforma	fuerte	débil	fuerte
fuentes	disponible	no disponible	disponible
POO	fuerte	fuerte	fuerte

Tabla 11 Comparación: PHP, ASP, JSP.

Veamos ahora algunos comparativos sintetizados de las características de ambos servidores.

	<i>IIS</i>	<i>Apache</i>
Peticiones		
Manejo independiente y múltiple de peticiones	SÍ	SÍ
Soporte de hilos	SÍ	SÍ
Lenguajes		
ASP	SÍ	SÍ, por medio de módulos ⁵⁴
Perl, Python, PHP	SÍ ⁵⁵	SÍ
JSP	SÍ	SÍ

Tabla 12 Comparación sintetizada. IIS, Apache.

54 Como Chilisoft de Sun o Apache::ASP.

55 Con la inserción de módulos. Lo mismo aplica para JSP.

Los tres manejadores de bases de datos, prometen ser los mejores por diversas razones, sin embargo siempre resulta conveniente consultar fuentes "neutrales" que nos permitan analizar las ventajas de cada opción de manera más crítica. Veamos las siguientes gráficas⁵⁶

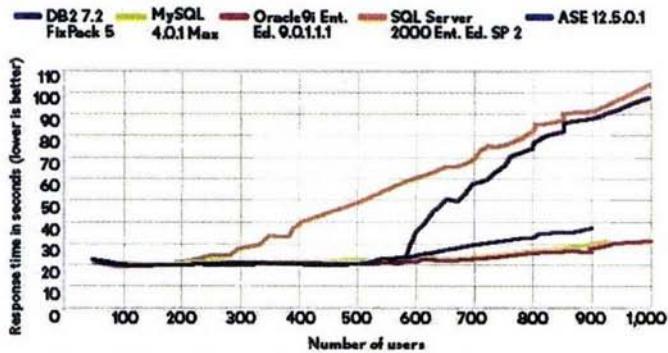


Figura 1 Tiempo de respuesta de DBA's respecto al número de usuarios

En esta primer gráfica, podemos observar que el rendimiento de los cinco manejadores de bases de datos es muy parecido hasta 200 usuarios. Es a partir de esta cantidad que el tiempo de respuesta se dispara en SQL Server mientras que los otros cuatro siguen relativamente constantes. Oracle y MySQL mantienen un rendimiento bastante bueno, su tiempo de espera se encuentra en un rango aproximado de (18, 32), por lo que el número de usuarios (al menos hasta 1,000) no es proporcional al tiempo de respuesta.

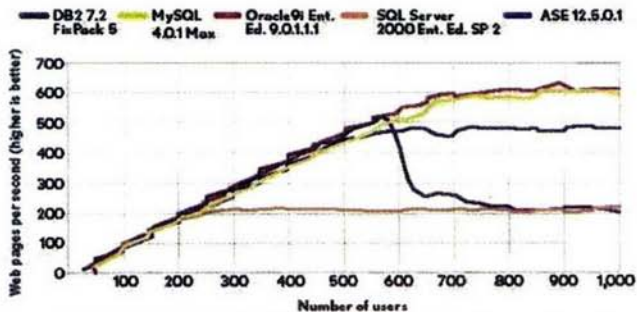


Figura 2 Páginas (con consulta a base de datos) por segundo respecto a número de usuarios.

56 Tomadas de la página www.eweek.com

Esta gráfica está orientada al análisis de respuesta de páginas web que dependen de bases de datos y refleja el número de páginas servidas con respecto al número de usuarios concurrentes. Se trata de un análisis de velocidad de conexión del servidor web con el servidor de base de datos. De nuevo, el rendimiento es muy similar con pocos usuarios (hasta 250 aproximadamente), momento en que la aplicación con SQL Server tiende a ser constante. En Oracle y MySQL esta tendencia se da con más de 600 usuarios triplicando además el número de páginas atendidas (de 200 a 600 aproximadamente).

Con base en este análisis, la mejor opción es la plataforma LAMP debido a su portabilidad (incluso con la posibilidad de pasar las aplicaciones a servidores IIS lo que no se puede hacer en sentido contrario). Existe la posibilidad de no gastar dinero a excepción del hardware aprovechando las licencias de los distintos paquetes (GNU es una de ellas).

Por otro lado, Linux es un sistema operativo bastante maduro en varias de sus distribuciones. Como se vió, los servicios tienen cierta independencia con el sistema operativo, por lo que de ser necesario, en cualquier momento se puede cambiar de distribución e incluso optar por sistemas "propietarios" como Solaris o MacOSX sin muchas dificultades.

Si bien, en muchos aspectos no existen diferencias destacables en las principales características de Apache y IIS, algo que es importante considerar es la prácticamente inexistencia de virus contra servidores Apache, los ataques a servidores IIS son más comunes debido al continuo descubrimiento de vulnerabilidades y la mala compartición de bibliotecas de su sistema operativo sede⁵⁷.

Si bien, el lenguaje más poderoso de los tratados es Java, para la complejidad y demanda del sistema tratado, es más que suficiente con PHP. Además de contar con una sintaxis sencilla de aprender, existen múltiples recursos de propósito específico (clases, procedimientos, etc.) disponibles en la red que pueden ser incluidos en el sistema⁵⁸.

Por el lado del manejador de bases de datos, MySQL ha demostrado tener muchas de las propiedades de otros manejadores como Oracle a un costo mucho más bajo. Al igual que Apache y PHP, se trata de software libre con un buen respaldo de su comunidad de desarrolladores y una buena cantidad de usuarios dispuestos a compartir sus experiencias y conocimiento sin recibir nada a cambio⁵⁹.

Por último, se debe tomar en cuenta que no sirve de mucho contar con el mejor servidor web, sistema operativo o manejador de bases de datos sin un administrador eficiente que esté siempre al tanto de la seguridad y correcto funcionamiento de las aplicaciones. Es necesario estar siempre al tanto de actualizaciones o vulnerabilidades además de revisar continuamente las bitácoras del propio sistema.

57 cabe señalar, que en torneos de crackeo de servidores web, realizar una intrusión a un servidor IIS otorga 1 punto, mientras que un servidor Apache, proporciona 5 puntos.

58 entre ellos, clases para envío de correos, formato de texto y ventanas de ayuda dinámicas.

59 algo de lo que más me agrada de la comunidad del software libre.

• CAPÍTULO 3

Desarrollo de la aplicación.

Cabe señalar, que en una etapa anterior, se ha realizado una primer aproximación del sistema de avisos totalmente dependiente de info. Esto lo señalo, para hacer notar el avance desde su primera concepción hasta la esperada al final del desarrollo. Como vimos en el primer capítulo, INFO es un shell script invocado desde */etc/aliases* y una clave apropiada que se encarga de enviar mensajes a través de un archivo de texto plano.

Precisamente, la versión "beta" de SAVII⁶⁰ consistía solamente de un área de texto para el mensaje (de nuevo en texto plano) y otra para el asunto. Lo único que realizaba el programa, era enviar el mensaje a *mensajeunix@pumas.iingen.unam.mx* concatenando la clave apropiada y este se encargaba del resto. Lo único que se logró, fue deshacerse de la dependencia con Outlook, pero había mucho por hacer.

Debo señalar que si bien, en ese momento tenía ya alguna experiencia en la creación de páginas web, fue mi primer contacto con PHP. ¿Para qué sirvió esta primera aplicación? En realidad para nada, nunca cumplió con su función y de hecho, jamás se envió un aviso real a través de él. Sin embargo, cumplió perfectamente su objetivo: Demostrar la viabilidad del proyecto. Fue con él que se realizó la propuesta al Jefe de Servidores UNIX.

3.1 Entrevistas con los usuarios.

3.1.1 Reunión con el jefe del Área de Servidores UNIX.

Cuando hablamos de SAVII nos referimos a un planteamiento inicial muy particular. Normalmente, un usuario tiene una necesidad, la que plantea al equipo de desarrollo y este se encarga de satisfacerla. En SAVII sí hubieron usuarios con una necesidad por satisfacer, pero lo especial, es que una buena parte de estos usuarios formarían el equipo de desarrollo. Se trata de un sistema desarrollado por los mismos usuarios⁶¹.

Todas la sesiones han sido planteadas sin un lineamiento, se trata simplemente de una plática sin estructura definida, que dé libertad de abordar las inquietudes más relevantes de cada una de las partes.

Como vimos al inicio de este apartado, la primera reunión, no propiamente una entrevista sino una lluvia de ideas, fue con el Jefe de ASU. Una de las grandes ventajas en las reuniones con él es que además de ser usuario, es Ingeniero en Computación y un programador experimentado, por lo que no existieron limitantes técnicas en la comunicación.

En esta primer plática, fue planteada la inquietud inicial de desarrollar un nuevo sistema que entre otras cosas, sea multiplataforma, pueda manejar texto con formato, maneje un sistema de

60 Que en este momento aún no había sido bautizado con ese nombre.

61 Una de las pretenciones, es que la gama de usuarios crezca más allá de la Coordinación de Cómputo.

colas automático y por si fuera poco que logre ser lo más independiente posible de nosotros.

La primer inquietud de ambos ha sido la seguridad. En INFO, el envío de correos era autorizado a través de una clave. ¿Cuál sería la mejor opción para SAVII? Luego de diversos planteamientos, se concluyó que el uso de una clave concatenada al asunto no sería la mejor opción, entre otras cosas debido al canal y a la dificultad de plantearle a un usuario "común" la necesidad de colocar una serie de caracteres extra al asunto que no serían incluidos en el envío. Es por todos conocido, que en muchas ocasiones (y esta no es la excepción), se debe procurar que para el usuario, los procesos sean lo más sencillos y acostumbrados posibles. La conclusión es simple, la mejor opción, es simplemente colocar una etapa de autenticación para acceder al sistema. La principal razón es que el usuario está muy familiarizado con ello, desde el acceso a su pc, acceso a su correo electrónico, etc.

Inmediatamente, salta una pregunta más: ¿Qué se utilizará como autenticación? La opción es el uso de un nombre y contraseña, pero hay tres combinaciones posibles:

- Las de acceso a la red.
- Las de acceso al correo electrónico (pumas.iingen.unam.mx).
- Una pareja independiente a las anteriores

Es esta una de las decisiones más complicadas debido a que las ventajas y desventajas dan cierto equilibrio. Entre ellas, junto al Jefe de ASU se han determinado las siguientes:

<i>Pareja de...</i>	<i>Ventajas</i>	<i>Desventajas</i>
Acceso a red	<ul style="list-style-type: none"> ✓ Todo el personal del II sabe de memoria la suya, pues la utiliza a diario para acceder a su equipo de cómputo. ✓ Se realiza a través de un servidor LDAP. 	<ul style="list-style-type: none"> x Depende del personal de Servidores Windows. x Si el servicio no está disponible, se cerraría la entrada al sistema aún si está funcionando de manera apropiada.
Correo electrónico	<ul style="list-style-type: none"> ✓ Se encuentran en el mismo servidor por lo que el acceso sería autónomo. ✓ Mayor control y discriminación de cuentas con acceso. 	<ul style="list-style-type: none"> x Debido al uso de clientes de correo capaces de guardar la contraseña, muchos usuarios no la saben
Independiente	<ul style="list-style-type: none"> ✓ No se comprometen las contraseñas del sistema. 	<ul style="list-style-type: none"> x Es un nombre de usuario y contraseña más por aprender

Tabla 13 Ventajas y desventajas de parejas para autenticación.

En base al análisis de estas ventajas y desventajas, se llegó a la conclusión de que la mejor opción es utilizar la pareja del correo electrónico, pero ahora surge otra necesidad: esta información no puede viajar transparente por la red, debe pasar por una etapa de cifrado. Sin embargo, este aspecto no es para preocuparse, pues gracias al capítulo anterior, sabemos que al servidor Apache se le puede integrar un módulo de cifrado por medio de SSL⁶² con lo que el problema queda totalmente resuelto.

Otra de las características solicitadas al sistema, es la necesidad de enviar texto con formato, imágenes y archivos adjuntos. Sin embargo, debido a la mayor complejidad y a la casi total ignorancia del tema se ha preferido dejarlo para otra ocasión.

Es con estos conceptos y luego de afinar los aspectos más relevantes, que el proyecto será presentado al Coordinador de Sistemas de Cómputo.

3.1.2 Reunión con el Coordinador de Sistemas de Cómputo.

En ésta ocasión, el esquema será un tanto más rígido, pues de esta reunión depende la realización o no del sistema. En una primera reunión, se intentó realizar el planteamiento general y la propuesta para que en una segunda, realmente tengamos una discusión sobre sus necesidades como un usuario más de la aplicación.

3.1.2.1 Primera reunión.

1. Planteamiento de la necesidad

Lo primero que se manifestó fue: "Quiero hacer un sistema de avisos nuevo" y la respuesta lógica (que de hecho ya se esperaba) llegó: ¿Para qué si ya existe uno?

Este es el momento de justificar. ¿Por qué vale la pena invertir tiempo y recursos en el desarrollo de un sistema de envío de correo informativo si ya existe uno que hace bien su trabajo?

El eje principal de la justificación es:

- ♦ Es una gran pérdida de tiempo para el personal de Sistemas de Cómputo, enviar correos informativos provenientes de todas las coordinaciones y secretarías.
- ♦ Se pierde tiempo en la espera de autorización por parte del coordinador para el envío de correo.

⁶² Protocolo diseñado por *Netscape Communications*, que permite cifrar una conexión y garantiza la autenticación. Se basa en criptografía asimétrica y en certificados. Funciona entre la capa de TCP y la de aplicación.

- ♦ En ocasiones, no deberíamos de enterarnos de cierta información contenida en los mensajes.
- ♦ Es necesario admitir texto con formato estándar.
- ♦ El envío por medio de clientes de correo es inseguro.

Es bajo estas cinco premisas que delimité los objetivos

Es siempre importante delimitar el alcance de una aplicación para determinar su viabilidad y lo que se espera lograr. Los objetivos básicos planteados para el proyecto son:

- ♦ Quitar de las funciones del Área de Servidores UNIX y del Coordinador de Sistemas de Cómputo el envío de mensajes informativos.
- ♦ Eliminar la dependencia de un cliente de correo electrónico o sistema operativo.
- ♦ Garantizar la privacidad de la información.
- ♦ Desarrollar un sistema de colas⁶³.
- ♦ Permitir la fácil integración de texto con formato e imágenes.
- ♦ Realizar un sistema de discriminación más detallado.

2. Viabilidad del proyecto

Basados en experiencias anteriores en el desarrollo de aplicaciones, era necesario antes que nada mostrar la viabilidad del proyecto, por lo cual fue mostrada la aplicación que envía mensajes sin formato.

Un aspecto a favor en la realización del sistema es el hecho de que además del pago de beca (recibida ya por el cumplimiento de funciones adicionales), no será necesaria la inversión de más recursos económicos. La razón es sencilla, el II cuenta ya con un servidor con las características requeridas (Apache + PHP + MySQL), no es necesaria la adquisición de software o hardware adicional.

3. Resultados de la reunión con el Jefe del ASU

En esta etapa, básicamente han sido planteados los conceptos surgidos de la reunión con el jefe de ASU. El objetivo, es presentar al equipo de desarrollo y los acuerdos tomados. Si bien, no se trata de papeles rígidos (a excepción del programador) se plantean tres identidades:

- ♦ Supervisor – Jefe de ASU
- ♦ Usuario – Coordinador de Sistemas de Cómputo
- ♦ Desarrollador – Yo

⁶³ Uno de los principales problemas de INFO, pues en caso de enviar dos mensajes de manera simultánea, el segundo sobrescribe al primero y este no se envía a todos los destinatarios.

Como ya fue señalado, las tres personas son a la vez usuarios y a través del desarrollo plantearán las necesidades que surjan en cualquier momento.

4. Señalamiento de requerimientos para el desarrollo.

Ahora, ¿qué se necesita para el desarrollo del proyecto?

Se cuenta ya con todas las direcciones de correo del personal académico y administrativo (heredadas de INFO), sin embargo, la mayoría de las cuentas de becarios se encuentran bajo servidores Windows que no dependen de ASU, por lo que se requiere que esa información sea proporcionada.

Por otro lado, si se ha planteado discriminar el envío de mensajes no solo por categoría sino por coordinación, subdirección e incluso edificio, se requiere reunir todos estos datos.

Es gracias a los primeros dos planteamientos, que se despertó el conocimiento de la necesidad. Con los últimos tres, se mostró que el proyecto es realizable y sus necesidades aceptables.

Ahora bastaba con esperar a la segunda reunión para saber si el proyecto sería realizado y con qué recursos se contaría.

3.1.2.2 Segunda reunión.

Esta reunión se realizó de nuevo sin un esquema definido. Solo se esperaba la decisión sobre la realización del proyecto y en su caso, el planteamiento de necesidades y recursos.

Sería el momento en que el "usuario" hablaría. Lo primero fue tal vez la mejor parte de la plática, el proyecto será realizado y dependerá únicamente del personal de Servidores UNIX⁶⁴.

¿Pero qué se necesitará además de lo planteado en la reunión inicial? Los señalamientos que se realizaron han sido precisos:

- Si se va a tratar de un mensaje con texto formateado e imágenes, se necesita un banner para todos los mensajes para otorgar identidad tanto al II como al propio sistema.
- Se requiere un registro de qué mensaje se envió, quién lo hizo y a quién fue dirigido.
- Es necesario tener una copia de cada mensaje enviado para evitar malentendidos o reclamos.
- El acceso restringido y seguro debe estar garantizado.

Afortunadamente, esto no fue todo lo tratado, en cuanto a las necesidades de desarrollo, se dieron algunas soluciones:

⁶⁴ Como vimos en el primer capítulo, existe dentro de la coordinación un área de desarrollo que en éste proyecto no tendrá participación.

- ♦ Las direcciones de correo de Servidores Windows, serán otorgadas.
- ♦ Para la obtención de los datos faltantes (coordinación, subdirección, ubicación), será enviado un mensaje solicitando la información a todas las personas en el II.

Con esto, el Coordinador de Sistemas de Cómputo, dio banderazo al proyecto y total libertad en cuanto al desarrollo.

3.1.3 Conclusiones de las reuniones.

Luego de la inquietud inicial de desarrollo del sistema y después de pasar por las reuniones, hay que señalar que la aceptación fue mejor que la esperada. Es curioso cómo en ocasiones una persona puede no saber de la necesidad de algo hasta que le es ofrecido y este es un caso de ello. En ambas reuniones (tomemos las dos con el coordinador como una sola), todos los planteamientos fueron aceptados por completo. El ofrecimiento de solución salvo algunos ajustes fue recibido totalmente.

Ante esta reacción, el compromiso de ambas partes, está de manifiesto. Si bien, el proyecto no era en sus inicios demasiado ambicioso y estaba planteado como una actividad secundaria, se ha convertido en objetivo primordial del área. Las condiciones están dadas.

Gracias a la metodología planteada, continuamente existirán pequeñas reuniones para la presentación de resultados, planteamiento de dudas, etc.

3.2 Diseño de la interfaz.

Si bien, parte importante del diseño es gráfico, no será abordado de manera profunda. Solo mencionaré que la gama de colores utilizada se ha elegido en base a la identidad del Instituto de Ingeniería (color rojo).

La distribución de página está basada en la división en dos frames. El superior sólo contiene un banner de referencia al sistema mientras toda la aplicación se concentra en el inferior.

3.2.1 Separación de etapas de la aplicación.

Basados en el análisis realizado, se han determinado las siguientes etapas:

- ◆ Autenticación

En la página de bienvenida, estará el módulo de autenticación completo, con respuesta en caso de negación de acceso. Existen dos opciones para la inserción de datos, una es hacerlo por medio de formas (simple código html), otra es realizarlo a través del navegador (con una ventana extra). La mejor opción será determinada luego de implementar ambas y comparar.

- ◆ Selección de gráficos

El banner para el encabezado de mensaje, puede ser seleccionado entre cuatro en base a las características del mensaje, existirán algunos con motivos festivos, formales o con detalles académicos, administrativos, etc. Se requiere de un método de selección exclusivo. Además, con el planteamiento de la existencia también de fondos dinámicos, es necesaria la selección de manera parecida a la anterior.

- ◆ Selección de destinatarios

Los factores de discriminación planteados de inicio son: Coordinación, Categoría y Subdirección. Los tres factores deben poder ser elegidos de manera independiente a los demás.

- ◆ Captura del mensaje

Como se trata de texto con formato, no basta con un "textarea" de código html. Es necesaria la implementación de botones parecidos a los que se encuentran en aplicaciones de oficina como Word. Como se señaló anteriormente, se trata de hacer un entorno familiar al usuario. Al ser un sistema nuevo para muchos de ellos, no debe ser demasiado especializado o complicado. El número de

caracteres horizontales será igual al de los editores de texto (generalmente 80).

- ◆ Envío del mensaje

Simplemente se trata de una etapa de confirmación. En ella, se presentará al usuario una vista previa del mensaje a enviar además de una descripción de los destinatarios para luego colocar la opción de modificación o envío.

- ◆ Confirmación del envío

Basta con una imagen o mensaje señalando que el mensaje fue enviado a todos los destinatarios de manera correcta.

Debido a sus características, he decidido conjuntar los módulos en páginas de la siguiente manera:

<i>Página</i>	<i>Módulos</i>	<i>Justificación</i>
Primera	Autenticación	No se debe presentar nada antes de que el usuario se autentique, basta con una presentación del sistema sin mayor información
Segunda	Selección de gráficos	Es necesario que las imágenes sean de buen tamaño para que el usuario pueda elegir su preferida. Si hablamos de cuatro banners y cuatro fondos, se requiere de todo el espacio en el navegador para desplegarlos con dimensiones apropiadas
Tercera	Selección de destinatarios Captura del mensaje	Una sola página puede contener ambos módulos. Dividiendo la página de manera vertical, el 70% será para la captura y edición del mensaje y el 30% para la selección de destinatarios.
Cuarta	Envío del mensaje	Aquí se integra el banner y el fondo con el mensaje desplegando ya la vista previa.
Quinta	Confirmación del envío	En una ventana final, se muestra simplemente si se envió correctamente o no el mensaje.

Tabla 14 Módulos del sistema.

3.2.2 Desarrollo de la interfaz⁶⁵

Con base en los requerimientos planteados anteriormente y con la continua comunicación y corrección de detalles, se llegó a los siguientes resultados⁶⁶.

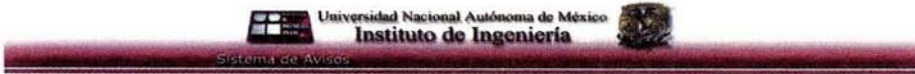


Figura 3 Banner del sistema

Para darle uniformidad al sistema, en el frame superior, solo se despliega este encabezado. El color rojo es característico del Instituto de Ingeniería. A la izquierda el logo del Instituto y a la derecha debido al interés de no disociarlo con la UNAM, siempre está presente su escudo.



Figura 4 Página de bienvenida

La página de bienvenida, cuenta con animación flash. El objetivo al colocarla es que cuando el usuario oprima el botón "Entrar", lo redirija a una dirección con cifrado y por medio de *htpasswd* reciba nombre de usuario y contraseña.

⁶⁵ ésta etapa fue realizada en conjunto con el Área de Desarrollo Web

⁶⁶ Imágenes adquiridas con Snapshot sobre los navegadores Mozilla y Konqueror



Figura 5 Selección de banner y fondo

En esta etapa, es necesario seleccionar el banner para el encajado del mensaje y el fondo.



Figura 6 La ventana principal de la aplicación

Aquí vemos la ventana de captura del mensaje y selección de destinatarios. La proporción trata de asemejarse a la de un cliente de correo común, el asunto sobre el cuerpo. La única diferencia es que debido a las diversas combinaciones de destinatarios, este menú se encuentra a la derecha.

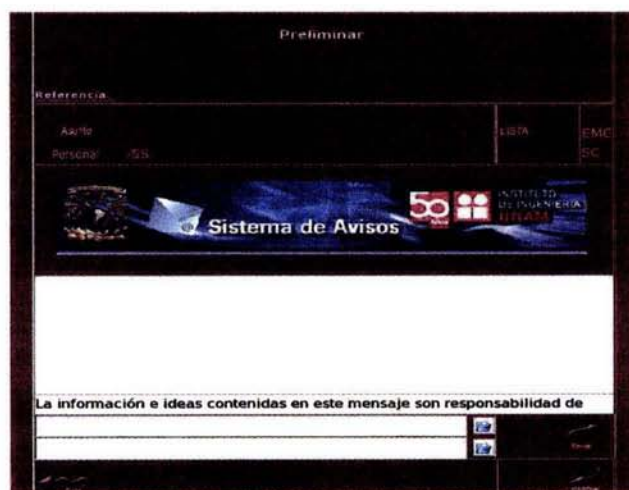


Figura 7 Vista preliminar

Antes de enviar el mensaje, se muestra cómo lo verán quienes lo reciban y a qué lista será enviado. De ser necesario, es posible volver al apartado anterior para realizar modificaciones.

Finalmente, el mensaje de confirmación de mensaje enviado exitosamente.



Figura 8 Mensaje enviado

3.3 Base de datos

La base de datos es tal vez la mayor diferencia (al menos al interior del proceso) con respecto a INFO que simplemente usaba un archivo de texto plano. En la base de datos, se contará con toda la información requerida para la discriminación de destinatarios.

3.3.1 Análisis de datos

Necesitamos contar con información de personal y con un registro de envíos. Los datos importantes para ambos casos son:

Nombre	Debido a la variedad de longitudes de nombre y al interés de separar por apellidos y nombres, se dividirá en tres campos
Título	Solo puede tener Pas, Ing, Lic, Mtro, Dr. Por ende, tendrá 4 caracteres.
Dirección de correo	También es variable, en promedio en el servidor local tiene 25 caracteres. Para evitar errores será puesto a 35.
Coordinación	Las coordinaciones tienen abreviaturas de máximo 3 caracteres.
Subdirección	Al igual que con las coordinaciones basta con 3 caracteres.
Categoría	ACA, ADM, BEC, SEC, SS, etc. 3 caracteres.
Puesto	Srio, Dir, Coord. Máximo se requiere de 5 caracteres.
Número telefónico	Debido a la probabilidad de que existan números de otras ciudades, se debe considerar la clave de larga distancia. P. ej. 0155 5622-69-63. Por lo tanto, 15 caracteres.
Cubículo	Puede ser numérico o alfanumérico (edificio 5 del II). Basta con 5 caracteres.
Edificio	Del 1 al 12 y la Torre de Ingeniería. Caracter de 2

Tabla 15 Información para la base de datos

3.3.2 Diseño de las tablas

De inicio, podemos considerar la necesidad de dos tablas: personal e historial. Por lo tanto, queda de la siguiente forma:

<i>Personal</i>		<i>historial</i>	
♦ Paterno	caracter 40	♦ Fecha y hora	caracter 16
♦ Materno	caracter 40	♦ Asunto	caracter 50
♦ Nombre	caracter 40	♦ Coordinación	caracter 5
♦ Título	caracter 5	♦ Subdirección	caracter 5
♦ Dirección de correo	caracter 30	♦ Remitente	caracter 40
♦ Coordinación	caracter 5	♦ Cuerpo del mensaje	caracter 40
♦ Subdirección	caracter 5		
♦ Categoría	caracter 3		
♦ Puesto	caracter 3		
♦ Número telefónico	caracter 20		
♦ Cubículo	caracter 5		
♦ Edificio	caracter 2		

Tabla 16 Tablas previas de la base de datos

Sin embargo, basados en solo dos tablas existe una gran repetición de información, lo que conlleva a ocupar más espacio en disco y hacer más lento el proceso. Es necesario *normalizar*.

Para ello, se pueden diseñar tablas solo de consulta como la de coordinaciones y subdirecciones y categoría⁶⁷.

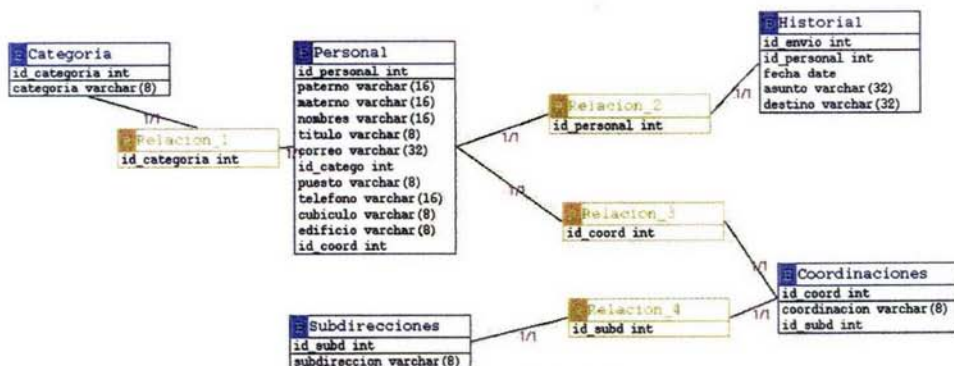


Figura 9 Diagrama entidad relación

67 Diseño de tablas y relaciones realizados con el programa Gerwin.

3.3.3 Implementación de la base

Existen varias formas de implementar una base de datos en MySQL. Una de ellas, es conectarse al servidor desde modo comando y por medio de instrucciones del tipo SQL.

```
create database nombre_base;  
use nombre_base;  
create table nombre_tabla (id int, nombre char.....  
.....
```

Sin embargo, existe una forma más sencilla de hacerlo por medio de herramientas como phpMyAdmin⁶⁸. Si bien el proceso es prácticamente el mismo, resulta mucho más fácil de realizar.

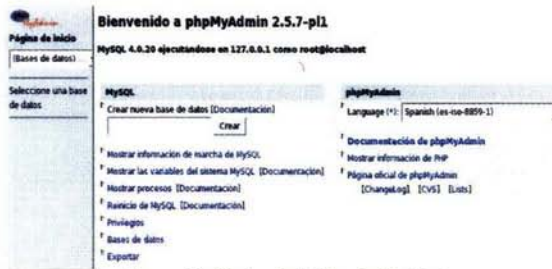


Figura 10 Página de inicio phpMyAdmin

Basta con poner el nombre de la base y crearla para luego generar cada una de las tablas⁶⁹. No es mi objetivo dar toda una reseña del proceso de creación de tablas y definición de índices. La única etapa que describiré a cierto detalle será la de determinación de datos, tipo y en su caso longitud.

⁶⁸ Se trata de una aplicación desarrollada en PHP para la administración de un servidor de bases de datos MySQL. Desde un entorno web, es posible por medio de sencillas operaciones crear tablas y bases de datos completas totalmente en modo gráfico.

⁶⁹ Incluso, Gerwin es capaz de generar código de MySQL que es posible introducir desde el cliente en modo texto o phpMyAdmin.

Database *sis_avisos* - Table *personal* running on 127.0.0.1

Field	Type [Documentation]	Length/Values*	Attributes	Null	Default**	Extra	Primary Index Unique --- Fulltext
id_personal	INT			not null		auto_increment	<input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
paterno	VARCHAR	30		not null			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
materno	VARCHAR	30		not null			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
nomibres	VARCHAR	30		not null			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
titulo	VARCHAR	4		not null			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
correo	VARCHAR	50		not null			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
id_coord	INT			not null			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
id_catego	INT			not null			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
puesto	VARCHAR	5		not null			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
telefono	VARCHAR	20		not null			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
cubiculo	VARCHAR	5		not null			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
edificio	VARCHAR	2		not null			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Table comments:

Table type:

Figura 11 Creación de la tabla *personal*

Es de destacar que en el campo *id_personal* caben propiedades como la de *auto_increment* que se encarga de que para cada registro, se incremente en uno el campo. Por otro lado, seleccionar *Unique* permite que este campo jamás se repita con lo que es único en todos los registros de la tabla.

Los campos *id_coord* e *id_catego* tienen exactamente las mismas características que los de las otras tablas (con las que serán relacionados).

3.4 Programación de módulos

El objetivo de éste apartado de ninguna manera es describir el proceso de programación de cada uno de los módulos. Simplemente quisiera señalar algunos de los momentos más interesantes, cuál fue el predicamento en algunos módulos y cómo fueron solucionados.

3.4.1 Autenticación

Para la etapa de autenticación existen muchos métodos (definitivamente todos ellos cifrados, lo que en realidad queda un poco ajeno a la aplicación pues se implementa en el servidor). La primer opción que me vino a la mente fue realizarlo con un simple archivo de texto plano con la estructura usuario {separador} contraseña, algo como esto:

```
alberto;Dkjff."$4  
victor;jdfhjh0345d
```

Bastaría con abrir el archivo con comandos de php (como `fopen`)⁷⁰. Sin embargo, aún con políticas de no indexación de directorios web, sería muy sencillo "robar" el archivo, digamos con un comando `wget`⁷¹ bien empleado, además de que no parece la solución más elegante. Definitivamente esta no es la solución.

Otra opción es generar otra tabla en nuestra base de datos del sistema o en otra base. Sin embargo, volvemos al mismo problema, hay que aplicar políticas muy estrictas para proteger correctamente la tabla (que no puede estar cifrada). Además, conlleva a programar la etapa de acceso.

La mejor opción no está en crear un archivo o una tabla. Es necesario voltear a las capacidades del servidor. Apache cuenta con comandos para generar archivos como el `/etc/shadow` del sistema. Incluso, es posible portar las líneas del propio `/etc/shadow` e insertarlas para limitar el acceso a un directorio determinado dentro de el servidor. El comando es:

```
dir_instalacion_apache/bin/htpasswd -c directorio_restringido/htpasswd usuario1  
dir_instalacion_apache/bin/htpasswd directorio_restringido/htpasswd usuario2
```

En la primera línea, se coloca el parámetro `-c` si no existe el archivo y debe ser creado. En la segunda ya no se pone el parámetro `-c` para que simplemente concatene la entrada al archivo ya existente. El resultado de estas operaciones es un archivo `.htpasswd`⁷² ubicado en el directorio

⁷⁰ En general, el manejo de archivos en PHP es muy parecido al de C.

⁷¹ Permite obtener archivos ubicados en un servidor web e incluso si se desea obtener todo un sitio (incluyendo subdirectorios). Es probable que en un servidor mal administrado, hasta se puedan robar archivos de configuración o administración como los ubicados en el directorio `/etc`.

⁷² Como el nombre del archivo empieza con `.` es oculto.

restringido. Un ejemplo de estos archivos es el que se da a continuación:

```
alberto:4L7UocQPJ2SmU:::
ebg:asvXcnTMkuMf:::
lch:4L7UocQPJ2SmU:::
```

Si se analiza, se observará que su estructura es muy parecida a la de `/etc/passwd`, y es que debemos recordar que anterior a la implementación de `/etc/shadow` para sacar la contraseña cifrada de `passwd`, la contraseña se encontraba precisamente como segundo campo en cada registro. Es por ello, que si se requiere un nombre de usuario y contraseña igual a los de una cuenta ya existente en el sistema, se puede acceder al archivo `/etc/passwd` para tomar el nombre de usuario y a `/etc/shadow` para tomar el bloque cifrado.

Lo único que falta por hacer es indicar a Apache que en este directorio se encuentra un archivo para autenticación, y esto se realiza con el archivo `.htaccess` colocado en el mismo directorio. Este archivo es de la siguiente forma:

```
AuthName "Sistema de Avisos "           //Etiqueta que tendrá la ventana de autenticación
AuthType Basic                           //Tipo de autenticación
AuthUserFile dir_apache/htdocs/dir/.htpasswd //Archivo donde se encuentran usuario y contraseña
AuthGroupFile /dev/null                  //Grupo con permisos de acceso

Options All
<Limit GET POST PUT>
require valid-user
</Limit>
```

Esto es todo lo necesario para realizar la función de autenticación. Al tratar de acceder a la aplicación, al usuario le será desplegada una ventana como la siguiente:

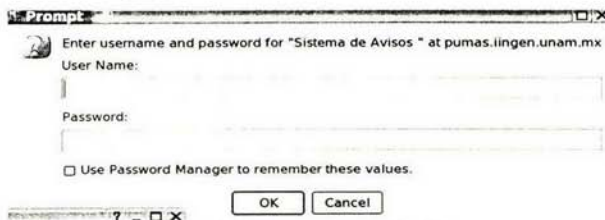


Figura 12 Ventana de autenticación

Otra ventaja, es la gran cantidad de opciones con que cuenta (dentro de los botones), una buena documentación y como en la mayoría del software libre, listas de ayuda de desarrolladores. Además, SPAW cuenta con versiones para ASP y PHP, con lo que simplemente, se toman las variables que genera y se manipulan al gusto.

Para integrarlo basta con colocarlo en un directorio dentro del *DocumentRoot* del servidor Apache y en la página que se desee colocar al inicio del archivo (dentro de etiquetas php), incluirlo de la siguiente forma:

```
$spaw_root = DR.'spaw/';  
include $spaw_root.'spaw_control.class.php';
```

donde *DR* es el *DocumentRoot* y *spaw* como es evidente, el directorio de *spaw*. Luego hay que incluir el archivo *spaw_control.class.php* como una biblioteca más de la aplicación. En donde se quiera colocar, simplemente se llama con

```
$sw = new SPAW_Wysiwyg('spaw1' /*name*/, "$content" /*value*/, 'es' /*idioma*/, " /*  
*toolbar*/, " /*theme*/, '95%' /*width*/);  
$sw->show();
```

en donde deben incluirse algunos parámetros como idioma o formato de la barra (ya que tiene varias formas de organización). La configuración se realiza en el archivo *spaw_control.config.php* (color, botones activos, etc).

Existe un inconveniente con SPAW: solo es compatible con Internet Explorer⁷⁶. Por ello, hubo la necesidad de implementar dos editores de texto enriquecido, uno para IE y otro para Mozilla. Con código PHP, se puede saber el navegador del cliente y en base a esa información, enviar uno u otro.

La opción de wysiwyg para Mozilla es RTE⁷⁷. La ventaja de este editor es que es multinavegador ya que soporta IE y Mozilla⁷⁸. Este es el punto en que se alcanza la multiplataforma.

¿Por qué no usar simplemente RTE y dejar de lado SPAW? Simple, desafortunadamente SPAW sigue teniendo más opciones y una mejor interfaz.

La integración con el sistema es de nuevo bastante sencilla, se editan las opciones en el archivo *richtext.js* y viene ya con un demo para unirlo al código PHP.

⁷⁶ Recordemos que uno de los objetivos del sistema es ser multiplataforma.

⁷⁷ Desarrollado por Kevin Roth.

⁷⁸ Todas las computadoras con Linux y algunas con Windows en el II lo tienen.

3.4.2.2 Envío del mensaje

Otra de las etapas complicadas es el proceso de envío del correo. La primera opción analizada fue utilizar el comando *mail* de Sendmail, sin embargo se trata de hacer una aplicación en cierta forma independiente del sistema operativo o Sendmail (al menos de manera directa). Otra opción era usar PEAR, un extensión de PHP que entre muchas otras cosas, cuenta con un módulo para envío de correo. Sin embargo, de nuevo hay que programar además de instalar la extensión.

Luego de una búsqueda, encontré la clase phpMailer. Es una clase programada con muchísimas opciones para envío de correo⁷⁹. De nuevo basta con colocarla en un directorio accesible para el sistema y mandarla llamar con los parámetros convenientes. En mi caso, es de la siguiente forma:

```
require "includes/class.phpmailer.php";
$mail = new phpmailer();
$mail->PluginDir = "includes/";
$mail->Mailer = "smtp";
$mail->Host = "pumas.iingen.unam.mx";
$mail->SMTPAuth = false;
$mail->From = "info@pumas.iingen.unam.mx";
$mail->FromName = "Sistema de Avisos";
$mail->Subject = $form[0]; // "Prueba de phpmailer";
$mail->Body = $cuerpo;
$mail->Timeout=620;
if ($adjunto1 != "none")
    $mail->AddAttachment($adjunto1,$adjunto1_name);
if ($adjunto2 != "none")
    $mail->AddAttachment($adjunto2,$adjunto2_name);
reset($direccion2);
while (list($clave, $valor)=each($direccion2))
{
    $mail->AddAddress($valor);
    $sexito = $mail->Send();
    $intentos=1;
    while((!$sexito)&&($intentos<2)&&($mail->ErrorInfo!="SMTP Error: Data not accepted"))
    {
        sleep(1);
        echo $mail->ErrorInfo;
        $sexito = $mail->Send();
        $intentos=$intentos+1;
    }
    if ($mail->ErrorInfo=="SMTP Error: Data not accepted")
        $sexito=true;
    if (!$sexito)
    {
        echo "Problemas enviando correo electrónico a ".$valor;
        echo "<br/>".$mail->ErrorInfo;
    }
}
```

⁷⁹ Desafortunadamente también perfecta para el envío de correo spam.


```

    }
    else
    {
        $mensaje=$valor;
        if ($archivo != "none")
        { }
        $mensaje.="</p>";
    }
    $mail->ClearAddresses();
}

```

A grandes rasgos, lo que observamos aquí es la configuración de dirección de origen, nombre de origen, destinatarios (viene de la selección), adjuntos, etc. Phpmailer se encarga de hacer el contacto con Sendmail (o algún otro servidor de correo como Qmail o Postfix) y enviar el mensaje.

3.4.3 Detección / Corrección de errores

Previo a la liberación del sistema, este ha sido expuesto a diversas pruebas para garantizar la inexistencia de errores⁸⁰.

La primer prueba fue la detección de errores en el módulo de autenticación. Básicamente se trataba de hacer combinaciones entre *login* (nombre de usuario) y *password* (contraseña). Veamos las posibilidades en base a una pequeña tabla de verdad⁸¹. Si el nombre de usuario es el correcto, es representado como 1, de lo contrario 0; lo mismo aplica para contraseña y salida:

<i>usuario</i>	<i>contraseña</i>	<i>salida</i>	<i>descripción</i>
0	0	0	Nombre de usuario y contraseña incorrectos: no permite acceso
0	1	0	Nombre de usuario incorrecto, contraseña correcta: no permite acceso
1	0	0	Nombre de usuario correcto, contraseña incorrecta: no permite acceso
1	1	1	Nombre de usuario y contraseña correctos: acceso permitido

Tabla 17 Tabla de verdad, usuario AND contraseña -> salida

⁸⁰ Estas pruebas fueron realizadas durante todo el proceso de desarrollo (tal como lo indica la metodología aplicada). Solo lo he puesto separado para dar orden al trabajo escrito.

⁸¹ Aunque tradicionalmente son utilizadas en diseño lógico, son muy útiles en otros ámbitos (este es un ejemplo de ello).

Esta fue la primer prueba importante y sin la cual, no era posible continuar con el desarrollo del sistema pues antes que nada, era necesario garantizar la restricción del sistema. Además de esta prueba en la etapa de acceso, fue necesario realizar un monitoreo a la comunicación para asegurarse de que la conexión se estaba realizando de manera cifrada⁸².

Por otro lado, también es importante asegurarse de que incluso con programas de obtención de archivos, no se puedan obtener ni los códigos ni los nombres de usuario y contraseña. Para realizar esta prueba, basta con utilizar el comando *wget* de la siguiente forma:

```
wget -c --no-parent http://direccion/subdir1/subdir2s/archivo

Resolviendo pumas.unam.mx... 132.248.53.245
Conectando con pumas.unam.mx[132.248.53.245]:80... conectado.
Petición HTTP enviada, esperando respuesta... 401 Authorization Required
Autorización denegada.
```

Al realizarlo, no permite obtener nada debido a que para acceder al directorio de la aplicación, se requiere de una autenticación. Si bien con esto es suficiente (tanto para garantizar que ni el código ni los archivos de autenticación sean tomados), realicé una prueba colocando los archivos de autenticación en un directorio no restringido:

```
wget -c --no-parent http://direccion/subdir1/htaces

Resolviendo pumas.unam.mx... 132.248.53.245
Conectando con pumas.unam.mx[132.248.53.245]:80... conectado.
Petición HTTP enviada, esperando respuesta... 403 Forbidden
14:29:56 ERROR 403: Forbidden.
```

Podemos observar que no lo encuentra. Por último y como una especie de demostración, copié uno de los archivos de la aplicación a este mismo directorio para ver si era posible extraerlo, veamos el resultado:

```
wget -c --no-parent http://pumas.unam.mx/archivo.php

--14:32:24-- http://pumas.unam.mx/archivo.php => `archivo.php'
Resolviendo pumas.iingen.unam.mx... 132.248.53.245
Conectando con pumas.unam.mx[132.248.53.245]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: no especificado [text/html]
[ <=>          ] 5,330  --.--K/s
14:32:24 (26.47 MB/s) - `archivo.php' guardado [5330]
```

Esta vez sí lo obtuvo, de esta manera podemos confiar en que todos los archivos están seguros.

82 Véase el apartado 4.1.2 Pruebas de seguridad para una breve descripción de las pruebas realizadas.

Siguiendo en el campo de la seguridad, podría ocurrir que alguna persona trate de acceder a un módulo antes de pasar por la etapa de autenticación o la selección de gráficos (y de hecho, en las pruebas se vio que ocurría). Es necesario establecer un mecanismo que en caso de que se trate de acceder directamente a alguna etapa, obligue al usuario a comenzar desde el principio.

La solución consiste en incluir en cada archivo la línea

```
include ("seguridad.php");
```

y en éste archivo, colocar:

```
<?
  if ($autenticado != "SI")
  {
    header("Location: https://pumas.iingen.unam.mx/Unix/infol/");
    exit();
  }
?>
```

\$autenticado es una bandera generada cuando el usuario se autentica e inicia desde la primer página. Así, si tratan de acceder a una dirección interna del sistema, es redirigido a la etapa de autenticación.

La siguiente etapa a probar es la de paso de variables a través de las diferentes páginas (tanto entre formas y PHP como entre PHP mismo). Básicamente las pruebas consistieron de asignar la variable en una página y asegurarse de su recepción en la siguiente de esta manera:

Página 1	Página 2
HTML	PHP
<input type="[text, radio, ...] name="variable" ...>	if (\$variable) echo \$variable;
PHP	
\$var="Valor esperado"	echo \$var;

Tabla 18 Paso de variables

En este caso, no hubo problema con la transferencia de variables (con sus respectivos

valores), el problema se vio cuando la variable era asignada en "Página 1" y debía ser usada no en "Página 2" sino en "Página 3". Al realizar las pruebas al respecto, se descubrió que en dicha página, la variable ya no existía.

Para solucionarlo, es necesario realizar un *registro de variable*. Resulta muy sencillo, si una variable generada en P1 será utilizada de nuevo en P3, simplemente se registra en P2 de ésta forma:

```
session_register('form'); session_register('marco');
```

De esta manera, no solo será vista la variable en todas las páginas, sino que su valor no cambiará (se convierte en una constante).

Una prueba vital fue la de formato del texto insertado en el mensaje. Simplemente se trató de enviar mensajes solo a mi cuenta con el mayor formato posible. El único error que SPAW reflejaba ocurrió en el uso de comillas pues al poner algo como "Hola", en el mensaje enviado ponía "\"Hola\""⁸³. Curiosamente, el error no era debido a algún error de programación sino a la misma configuración de PHP. Luego de una breve investigación, descubrí que en *php.ini* existe la regla

```
; Magic quotes for incoming GET/POST/Cookie data.  
magic_quotes_gpc = On
```

que hay que pasar a

```
magic_quotes_gpc = Off
```

Finalmente, tal vez la prueba más importante: "carga". Es necesario, evaluar la capacidad del sistema en cuanto a envío de correo masivo (más de 500 direcciones). Lógicamente, no se podía realizar con las direcciones reales de los usuarios, así que simplemente, creamos una tabla temporal con un bloque de nuestras direcciones de correo tanto de servidores comerciales como los propios del II. De ésta manera, la base de datos quedó conformada así:

dir1@yahoo.com	100 veces
dir2@yahoo.com	100 veces
dir3@yahoo.com.mx	100 veces
dir4@hotmail.com	100 veces
dir5@pumas.iingen.unam.mx	200 veces
dir 6@pumas.iingen.unam.mx	200 veces
dirfalsa@servidorfalso.com	50 veces

Existían varios objetivos:

⁸³ comúnmente se conoce como "escapar" las comillas

- ♦ Comprobar que ante la carga de envío, el servidor no "se sentaría"
- ♦ Ver la respuesta ante direcciones inexistentes
- ♦ Observar la respuesta ante el envío a cuentas de correo sobre la cuota del servidor (en el caso de yahoo y hotmail)
- ♦ Determinar el tiempo de tardanza.

El primer resultado fue desalentador, el envío tardó alrededor de 3 minutos pues el proceso se vió truncado, y solo la mitad de los mensajes esperados llegó al destinatario. Se bajó el tiempo de espera ante error en el destinatario de cinco a un segundo y la cantidad de intentos a una misma dirección de 4 a 2⁸⁴.

El resultado fue prácticamente el mismo, de nuevo tardó 3 minutos, y no fueron entregados todos los mensajes. El problema estaba de nuevo fuera del programa, se encontraba en la configuración de PHP.

La solución que se dió fue aumentar los valores de las siguientes variables en *php.ini*:

```
max_execution_time = 300 ; Maximum execution time of each script, in seconds  
max_input_time = 60
```

Luego de estas modificaciones, el sistema no volvió a presentar problemas en esta etapa.

⁸⁴ Todo ello con una simple modificación del código.

CAPÍTULO 4

Implementación y liberación.

Previo a la liberación, son necesarias algunas pruebas generales. Para este caso, debido a que el equipo de desarrollo mismo era parte de los usuarios, muchas de las pruebas "de usuario" han sido realizadas por nosotros mismos (antes de liberarlo en forma).

4.1 Pruebas internas.

4.1.1 Pruebas de formato.

Básicamente, lo que hice fue mandarme mensajes solo a mí (para ello, aproveché que de momento no existe personal de Servicio Social en la CSC y me asigné esa categoría) con todas las posibilidades en cuanto a color, alineación, estilo, etc.

Como ya lo había notado durante la etapa de pruebas, el único error era el hecho de que las comillas se reflejaban en el mensaje como \"`escapadas`\", por lo que simplemente fue necesario configurar PHP para que esto no ocurriera.

Un error que fue detectado por otra de las personas que hicieron las pruebas internas (el jefe de área) fue que el mensaje tenía una longitud de renglón determinado en el sistema (65) y al ver el mensaje ya recibido, no se mantenía constante.

Por las características de MIME y el hipertexto, sabemos que la longitud de los renglones depende del tamaño del "contenedor" en que se encuentran, es decir si el texto se encuentra en una tabla o celda de 50 caracteres por renglón, el texto se adaptará a las dimensiones y los párrafos solo tendrán 50 caracteres por renglón.

Por ello, la única "solución" que dí, fue que el número de caracteres del renglón al interior del sistema fuera de 80 (el estándar desde las máquinas de escribir). De esta manera, se puede homogeneizar un poco la manera en que se verá el mensaje. El problema está en que depende del cliente de correo usado la forma en que se verá el mensaje, ya sea Outlook, Eudora o IMP (cliente por página web) y ya no decir pine (que algunos siguen usando y no es capaz de desplegar gráficos).

Desafortunadamente, fuera de ésta aproximación, el problema queda fuera del alcance del sistema.

4.1.2 Pruebas de seguridad

Antes que nada, es necesario considerar las tres características básicas de la seguridad: *disponibilidad, confidencialidad e integridad*.

Como breve resumen podemos señalar que la disponibilidad consiste en que el recurso esté al alcance del usuario siempre que este lo requiera; confidencialidad es que la información no sea conocida por ninguna persona que no deba hacerlo; integridad es que la información permanezca sin ningún cambio no deseado⁸⁵.

La disponibilidad está garantizada al encontrarse el sistema en el servidor web y de correo electrónico del Instituto de Ingeniería. Cuenta con conexión directa al IIMAS (puente hacia DGSCA) por lo que incluso cuando no hay servicio de red al interior del II, el servidor se mantiene disponible al exterior. También se utiliza el mismo servicio de web que para las páginas alojadas y el servidor de base de datos. Como prueba basta señalar que este servidor solo ha sido apagado en 4 ocasiones durante los últimos dos años (ya sea debido a mantenimiento a la red eléctrica o por remodelación del área física en la que se encuentra) y ha sido durante periodos muy cortos.

La confidencialidad está garantizada gracias al módulo de SSL del servidor Apache. Además, la información de la base de datos se encuentra resguardada pues ninguna aplicación que no se encuentre en el propio servidor tiene acceso a ella además de que para el acceso a la tabla de personal se requiere de nombre de usuario y contraseña. Para ello, se trató de acceder a la base de datos sin el uso del sistema o las contraseñas y no fue posible obtener la información.

También ha sido escaneada la conexión con el servidor por medio de los programas ethereal y ntop⁸⁶. Ninguna información es observada a través de ellos.

El caso de la integridad es un poco más complicado. Al ser un sistema "pequeño", no tiene mucho caso implementar mecanismos rebuscados para garantizarla⁸⁷. Para este caso, las pruebas no se hicieron tanto respecto a la veracidad de la información en sí, sino al hecho de que al personal elegido para el envío sea al único que le llegue el mensaje. En este caso, los resultados fueron satisfactorios a excepción del caso en que se seleccionaba a los subdirectores, pues no enviaba nada. Para ser sinceros, se trató de un error del programador pues coloqué un comentario en donde se hacía ésta discriminación. Todo quedó solucionado al quitar dicho comentario.

85 Solo se trata de una reseña breve, se puede ahondar en el tema en diversos libros de criptografía y seguridad informática.

86 Ambos software libre incluidos en la distribución SuSE Linux 9.1.

87 Solo existe un itinerario regular de respaldo de las tablas.

4.1.3 Pruebas de carga (desempeño)

Ya con direcciones reales de los dos servidores de correo del II (e incluso servidor UNAM y algunas cuentas comerciales), se realizaron las primeras pruebas. Ahora, la distribución de sujetos de prueba (que ya son los reales) es:

por Categoría		por Subdirección	
Académicos	180	Estructuras	290
Administrativos	90	Ambiental	260
Becarios	550	Elec-mec. y cómputo	190

por Servidor	
@pumas.iingen.unam.mx	266
@iingen.unam.mx	500

Tabla 19 Distribución de personal para envío de mensajes

De estas tablas podemos observar que la mayor carga ocurre cuando se envían mensajes a la Subdirección de Estructuras o cuando se envía mensajes a los becarios de todo el II (dejando de lado por un momento el envío a todo el personal sin importar categoría o coordinación). Por lo tanto, los primeros envíos han sido realizados primordialmente a académicos de las subdirecciones de Ambiental y Electromecánica y Cómputo.

En los primeros mensajes, continuamente han existido errores, simplemente después de pasado un tiempo, el sistema para. Al observar las bitácoras de PHP, no se observó nada, tampoco en las de Sendmail. Por lo tanto, es un error de programación. Un aspecto que pasó desapercibido desde su implementación fue que la clase phpMailer cuenta con un tiempo límite para el envío de mensajes que era de 100. Simplemente se elevó esta cantidad y no siguió existiendo el problema

4.1.4 Pruebas de contingencia

Como se vio en el capítulo anterior, ocurrió en las pruebas de desempeño que el sistema se quedaba "pasmado" antes de enviar la totalidad de los mensajes. Si bien, el defecto fue reparado, nunca está de más prevenir. ¿Qué hacer en caso de emergencia?

Tal vez no sea la manera más elegante, pero la siguiente es efectiva. Al momento de generar la lista temporal de direcciones, esta es copiada a un archivo de texto que es guardado en un directorio con el historial de envíos, luego al ir mandando los mensajes, se concatena cada dirección a otro archivo denominado *enviados*. De esta manera, en caso de error, basta tomar los archivos y volver a enviar solo los que no lo fueron desde la primera vez.

Para diferenciar cada uno de los archivos, a su nombre se concatenan fecha, hora e identificador de la sesión PHP, lo que garantiza que ninguno de ellos se sobrescriba. También existe el apoyo de la tabla historial que como su nombre lo indica, lleva un registro de los envíos de avisos a través de SAVII.

4.2 Implementación.

La implementación en cuanto a puesta en funcionamiento en el servidor fue muy sencilla debido a que todo el trabajo fue desarrollado sobre el servidor en producción. Simplemente hubo que darlo a conocer y dar de alta a los usuarios correspondientes.

4.2.1 Etapas de liberación.

De nuevo, es necesario recalcar un aspecto particular en este sistema: algunos de los usuarios forman parte del equipo de desarrollo, lo que debe tomarse en cuenta en las etapas de liberación.

Desde que existió el módulo de envío de mensajes (incluso todavía sin formato) y la base de datos estuvo completa, se buscó enviar algunos de los mensajes que nos eran solicitados a través del "minisistema". Los únicos usuarios partícipes en esta etapa fueron el Jefe de Área de Servidores UNIX y yo mismo. Durante esta etapa, no existía aún el módulo de autenticación por lo que se tomaba el riesgo de habilitar el directorio y archivos del sistema durante la prueba y luego deshabilitarlos de nuevo. Ni siquiera eran enviados con banner los mensajes.

La que puede ser considerada segunda etapa sería cuando el módulo de autenticación (todavía basado en un archivo de texto) fue implementado⁸⁸.

Para la tercera etapa, ya se contaba con los banners de encabezado del mensaje.

Ya para la cuarta, se contó con texto con formato (primero solo para IE y después para Mozilla), pero aún no daba los resultados esperados en las diferentes pruebas.

La liberación final se dio cuando todas las pruebas fueron satisfactorias y ciertamente no fue complicada. Si consideramos que el desarrollo fue realizado sobre el servidor en producción y justo en la ruta deseada⁸⁹, simplemente fue necesario colocar todas las cuentas en el archivo para acceso y darlo a conocer.

4.2.2 Capacitación

En mi primer empleo como programador uno de los usuarios de nuestros sistemas me dio una buena lección. Durante la capacitación que se le ofreció en el uso de nuestro sistema (que varios de sus empleados utilizarían), mandó llamar a quienes él consideraba menos "entendidos", por decirlo de alguna forma, en el uso de este tipo de aplicaciones. Comentaba en principio, que si ellos aprendían a usarlo, cualquier otro lo podría hacer. Este hecho en conjunto a lo aprendido en alguna de las clases de

⁸⁸ fue justo aquí cuando se vió la necesidad de utilizar al propio servidor como forma de autenticación

⁸⁹ sencillamente, estaba restringido para algunas cuentas

licenciatura en la que continuamente se señalaba "el mejor programa está hecho para tontos"⁹⁰ me llevaron a hacer un sistema lo más sencillo posible.

La capacitación ha sido totalmente personalizada en una breve sesión de menos de una hora en la que se explica al usuario cómo instalar el certificado de seguridad y cómo elegir el banner (en conjunto con algunas recomendaciones de colores y formato).

La parte medular es la captura del mensaje y la selección de destinatarios. Gracias a que el entorno tiene características muy similares a las de programas de edición de texto como Word, la aceptación y aprendizaje son inmediatos. Solo se ha recalcado la etapa de selección de destinatarios señalando todas las combinaciones posibles y los resultados que deben esperarse.

En lo que más se insistió fue en el hecho de no cerrar el navegador hasta que la página de confirmación de envío sea desplegada⁹¹.

90 Mtro. Ernesto Peñalosa Romero

91 Para mayor información, ver el Apéndice 1. Manual de usuario.

4.3 Modificaciones subsecuentes

Como era de esperarse, algunas cosas tuvieron que ser modificadas a petición de los usuarios, señalaré a continuación algunas de las más interesantes.

La primera de ellas, que fue realizada prácticamente al interior del área fue el planteamiento de la necesidad de que el sistema funcionara exactamente igual en Internet Explorer que en Mozilla. La razón por la que no se procuró este que era uno de los objetivos iniciales del proyecto, fue el estimado de que el noventa por ciento de los usuarios del sistema contarían solamente con el primer navegador, por lo que se podía dejar este aspecto en segundo plano.

Sin embargo, ya funcionando la aplicación era necesario cumplirlo. Para ello como ya se señaló, se implementó SPAW para Windows+IE y RTE para cualquier SO+Mozilla.

Esta etapa es fiel prueba del funcionamiento de la metodología utilizada, pues en cuanto estuvo listo, el usuario de inmediato pidió que la superficie de captura del mensaje fuera reducida pues al contrario de la de IE, se veía mal proporcionada y hasta poco estética.

Tomando en consideración que no todos los mensajes serán enviados por la persona responsable de la información contenida en ellos, se solicitó el contar con alguna forma de desafanar al encargado de enviar el mensaje del contenido de este.

Para ello, en la etapa de captura de mensaje (entre asunto y cuerpo del mensaje), se colocó otra entrada para colocar el nombre del responsable del mensaje, que sería reflejada al final de él. Aunado a este detalle, todos los mensajes tienen como pie de página el nombre de la persona que envió tal mensaje.

Otra petición se dió en el caso de la selección de personal por categorías, pues un usuario más solicitó alguna manera de enviar el mensaje a todas las categorías sin la necesidad de seleccionar cada una de ellas por separado. Por ello se colocó un botón "Seleccionar todos" que ahorra este proceso (en complemento con el botón "Quitar selección" que como es lógico, quita todas las selecciones).

Como se puede apreciar, la mayoría de las modificaciones solicitadas han sido solamente de formato (básicamente estéticas). Otro ejemplo de ello, es que en la vista preliminar del mensaje, se veía el banner con el fondo rojo propio de la página, cosa que al Coordinador de Sistemas de Cómputo no le agradó mucho, por lo que pidió que se viera tal como sería el mensaje enviado (si está incrustado sobre un fondo blanco, así debe ser desplegado en la vista previa). Bastó con cambiar el color de fondo de esa celda de la tabla.

Al existir un error en alguno de los mensajes enviados (destinatario individual), el programa estaba originalmente diseñado para esperar 5 segundos e intentar enviarlo de nuevo (en total en tres ocasiones). Como era de esperarse, esto retrasaba el proceso total (incluso a varios minutos). A

solicitud de varios usuarios, se disminuyó a 3 segundos. No conformes con el avance, se bajó a solo 1 segundo y 1 intento extra. De nuevo, no quedaron conformes con el rendimiento, con lo que finalmente se tuvo que eliminar esta redundancia.

Tal vez una de las solicitudes más complejas fue la inclusión de una base de datos especial para un usuario del sistema perteneciente al Consejo Universitario que deseaba enviar mensajes a correos totalmente distintos a los encontrados en la base estándar. Como también requería tener acceso a la propia base para modificarla, se optó por crear una nueva tabla que solo es accesible por el sistema cuando el que accede es él.

Como integrante del grupo de usuarios, yo mismo planteé la necesidad de una asistencia sin la necesidad de trasladarse a otra página o revisar un manual completo⁹². De esa forma, se incluyó la biblioteca Overlib⁹³ desarrollada en Javascript que despliega ventanas de información sobre la propia página.

Otro aspecto que de inicio quedó en segundo término sería la capacidad de envío de archivos adjuntos (solicitada en una de las pruebas en la que se tuvo que usar INFO por la carencia de esta capacidad). Se optó por colocar este módulo en la etapa de confirmación para evitar tiempos muertos en caso de requerir modificar el mensaje. Se determinó la suficiencia en dos archivos adjuntos como máximo.

Alguna de las últimas solicitudes realizadas sería la de un módulo accesible solo para los administradores (Coordinador de Sistemas de Cómputo, Jefe de ASU y yo) en el que simplemente se vea el registro de los mensajes enviados⁹⁴.

Para que sea lo más dinámico posible, el campo de fecha de la tabla *historial* ha tenido que ser dividido en cuatro campos: año, mes, día y hora. De esta manera, la selección puede ser mucho más detallada en el momento de la consulta.

El módulo de administración presenta un menú por año, mes, día y remitente además de determinar el orden de acuerdo a fecha, asunto y remitente.

92 La idea la tomé de la página de inscripciones al posgrado de la UNAM.

93 Desarrollada por Erick Bosrup.

94 Como aportación personal en éste punto, incluí la posibilidad de generar un archivo de formato PDF con los envíos seleccionados.



Conclusiones

En primera instancia, puedo señalar que la aplicación de la Programación Extrema ha sido una excelente opción en el desarrollo de este tipo de aplicaciones (tanto por su complejidad como por la composición del equipo de desarrollo) debido a la integración del equipo de trabajo con responsabilidades variables, pero sobre todo la interacción directa y constante con el usuario. Gracias a ella, el desarrollo es más rápido y permite que el tiempo de liberación sea mucho menor pues en todo momento, el usuario comenta los cambios que desea y se conocen los fallos del sistema. En otras metodologías, buena parte de estos aspectos, se ven hasta que el usuario ve por primera vez el sistema completo y los desarrolladores requieren volver a etapas cubiertas tiempo atrás para resolverlo.

Las características de PHP han resultado excelentes para este desarrollo. Primero se puede mencionar que su sintaxis es en verdad muy parecida a la de C con lo que la curva de aprendizaje fue bastante pequeña. A diferencia de C sin embargo, cuenta con muchas funciones ya programadas que simplifican la programación. Gracias a su integración completa con el servidor Apache, la velocidad de envío ha sido bastante buena. MySQL también ha demostrado su gran velocidad. El único "participante" que dejó que desear fue Sendmail que en momentos de mucha demanda, reflejaba cierta lentitud y hasta carencia de recursos.

El tiempo de desarrollo se vio abatido debido a la adquisición de varios de los módulos a través de internet gracias a programadores experimentados dispuestos a compartir sus desarrollos.

Si bien el tiempo activo de desarrollo fue breve, en tiempo real, fue demasiado largo. Siempre que se plantee un desarrollo de cualquier nivel de complejidad es necesario establecer tiempos apropiados pero sobre todo cumplirlos para evitar retrasos.

Aprovechando la riqueza de la base de datos, próximamente se desarrollará una agenda con módulos de búsqueda, envío de información al correo electrónico de quien consulta y creación de archivos PDF.

Si bien, durante el desarrollo se fueron cumpliendo los objetivos y ya para su liberación SAVII superó a INFO, faltan algunos puntos a cubrir entre los que se puede mencionar: validación de información enviada, respuesta automática ante errores en envíos y lo que a mí en particular me parece más complejo (pues a mi parecer, las características de PHP y Apache no lo permiten), enviar la petición de envío al servidor y cortar la conexión sin que conlleve al término del proceso de envío sin la finalización satisfactoria. Otra necesidad es la creación de perfiles de usuario dinámicos y sencillos de implementar que no conlleven a un cambio drástico en el código (probablemente con una base de datos o con POO).

APÉNDICE 1

Manual de usuario.

UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

INSTITUTO DE INGENIERÍA

Coordinación de Sistemas de Cómputo

Área de Servidores UNIX

SAVII 1.0

Manual de usuario.

Alberto Barrón Cedeño
alberto@pumas.iingen.unam.mx
562280-92 al 97

agosto, 2004

1. Objetivo

El objetivo principal de SAVII, es independizar el envío de mensajes de corte informativo a la comunidad del Instituto de Ingeniería de la Coordinación de Sistemas de Cómputo. Se trata de una interfaz accesible en Internet, que permite luego de un pertinente loggeo enviar mensajes al personal basado en categorías, coordinación, subdirección y puesto.

2. Requerimientos en el cliente

- Conexión a Internet
- Internet Explorer o Mozilla

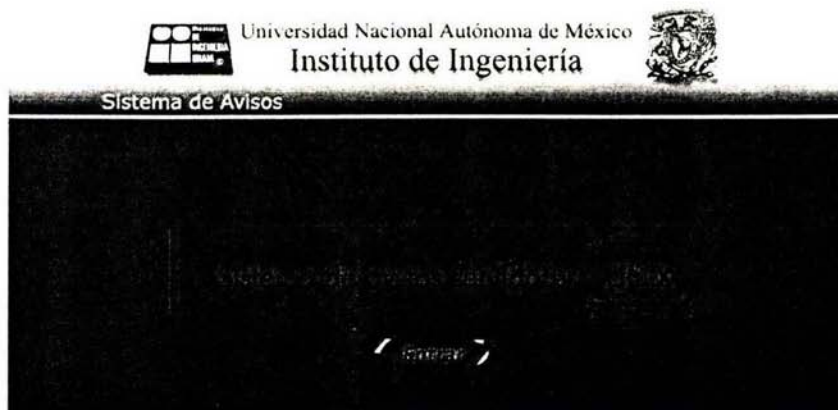
3. Descripción del sistema

3.1 Acceso al sistema.

Para entrar a SAVII abra su navegador de Internet (IE/Mozilla) y acceda a la siguiente dirección:

<http://pumas.iingen.unam.mx/Unix/info/>

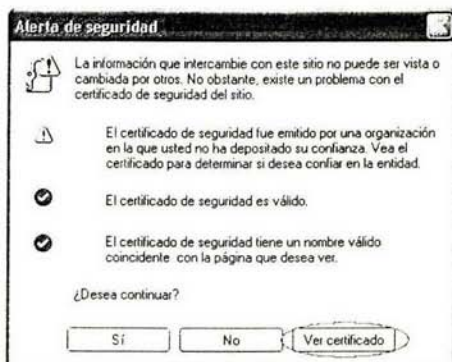
se encontrará con una página de bienvenida. Oprima la imagen de entrar



Para garantizar la seguridad de su información (y esta no pueda ser vista a través de la red), SAVII cuenta con un certificado de seguridad. En la primera ocasión que entre a la aplicación, deberá

instalar éste certificado en su ordenador de la siguiente forma:

Oprima el botón **Ver Certificado**



Esta acción despliega el certificado con algunos datos acerca de este.

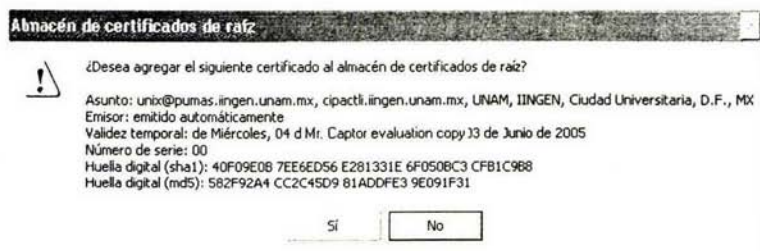
De click en el botón **Instalar Certificado...**

Y oprima **Siguiente** en la próxima ventana.





Aparecerán varias ventanas en las que solo debe oprimir **Siguiente** hasta que aparezca esta:



Elija la opción **Si** y en la siguiente ventana **Aceptar**



Por último, presione el botón **Si y listo**. El certificado está instalado. En próximas ocasiones, no tendrá que seguir este proceso.

Ahora aparecerá una ventana de autenticación. Basta con colocar el nombre de usuario y contraseña que le fue proporcionado (los mismos que en el correo electrónico de pumas)

Enter username and password for "Sistema de Avisos " at pumas.iingen.unam.mx

User Name:
alberto

Password:

Use Password Manager to remember these values.

OK Cancel

Al proporcionar los datos correctos, se le dará acceso a la aplicación.

3.2 Selección de gráficos para el mensaje

En la siguiente pantalla, observará el menú de selección para el banner y el fondo que desea en su mensaje.



El banner por defecto es el de color azul que hace referencia al 50 aniversario del Instituto de Ingeniería. Para provocar un sentido de identidad al sistema y al mismo Instituto, es recomendable siempre usar esta imagen. El fondo por defecto es el blanco liso. Ambos aparecen seleccionados desde el momento de entrar.

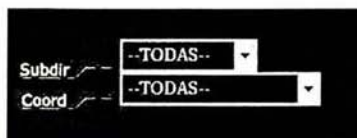
Luego de hacer su selección, oprima el botón continuar

3.3 Escritura del mensaje y selección de destinatarios.

La siguiente página, es la más completa de todas. Lo que verá en la parte superior es el lugar en el que pondrá el asunto que desea para su mensaje además de el responsable del mismo.

Asunto	Aviso. Instituto de Ingeniería
Responsable	Alberto Barron

A la derecha, se encuentra el menú de Subdirección y Coordinación. La opción por defecto es enviar el mensaje a todas las subdirecciones (y por ende todas las secretarías y coordinaciones). De no requerirlo así, seleccione la subdirección a la que desee enviar el mensaje o incluso la coordinación que requiere.



Recuerde, el Instituto de Ingeniería esta organizado por las siguientes subdirecciones y coordinaciones.

Ambiental

Electromecánica

Estructuras

y Cómputo (EMC)

Hidráulica

Automatización

Estructuras y Materiales

Ingeniería Ambiental

Instrumentación

Geotecnia

Bioprocesos Ambientales

Mecánica, térmica y fluidos

Mecánica aplicada

Ingeniería de Procesos

Ingeniería de Sistemas

Sismología

Sistemas de Cómputo

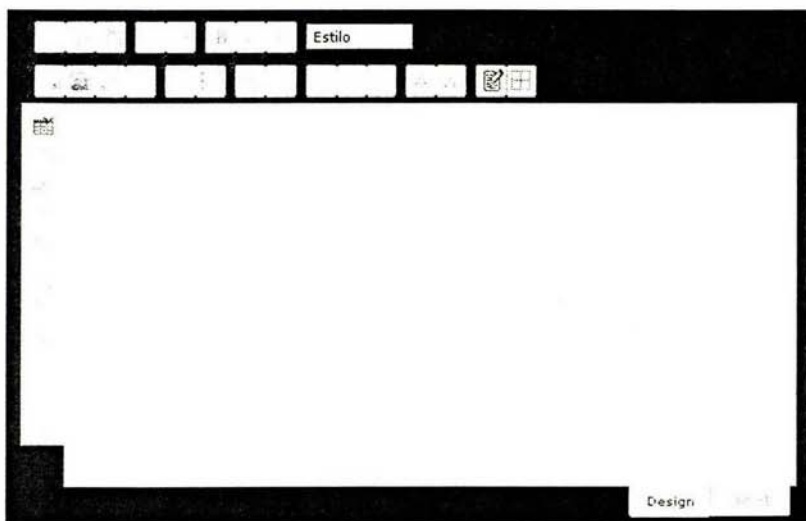
Vías Terrestres

Ingeniería Sismológica











Debajo de este menú, se encuentra el área de selección de tipo de personal. Si su mensaje debe ir dirigido a todos, simplemente oprima el botón de "Seleccionar todos", de lo contrario, haga su selección uno por uno.

En la celda principal de la página cuya descripción se ha dejado al final, se encuentra el área en la que usted escribirá el mensaje.



Este módulo, es muy parecido al usado en su editor de texto (Word por ejemplo). A la izquierda están los botones para creación y manipulación de tablas. En la parte superior e izquierda a derecha están los botones:

<i>Botón</i>	<i>Función</i>	<i>Descripción</i>
	Cortar	Seleccione un texto y oprima este botón para cortarlo. Estará disponible si desea pegarlo en alguna otra posición.
	Copiar	Seleccione un texto y oprímalo para copiarlo.
	Pegar	Pegará un texto cortado o pegado previamente (dentro o fuera de esta aplicación).
	Deshacer	Deshace la última operación realizada (escribir, cortar, borrar, etc).
	Rehacer	Vuelve a hacer la ultima operación realizada.
	Negrita	Seleccione un texto y oprima para asignarle el estilo negrita. Las opciones de itálica y subrayada funcionan de la misma forma.
	Itálica (cursiva)	
	Subrayada	

Botón **Función**



Enlace

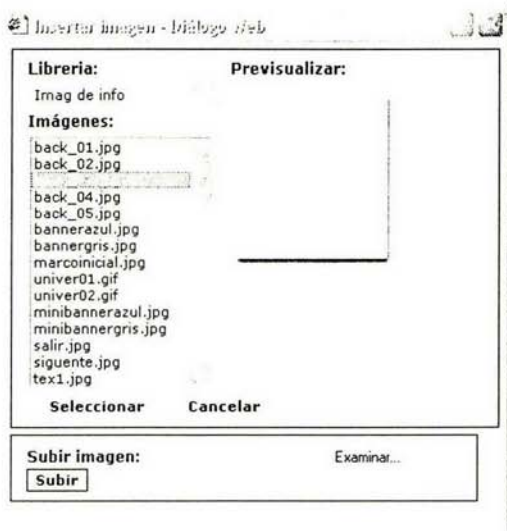
Descripción


Si desea incluir un enlace a una dirección URL, seleccione el texto que desee se vea afectado y oprima este botón. En la ventana que aparece, escriba la dirección en el campo URL. Si lo desea, puede utilizar otros protocolos o funciones. Debe seleccionarlos en tipo.

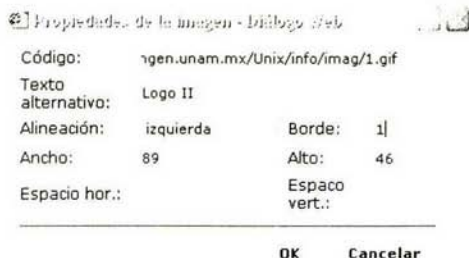









Insertar imagen



Si desea insertar una imagen en el cuerpo del mensaje, elija esta opción. De nuevo, se abre una ventana en la que podrá elegir una imagen de las disponibles en la biblioteca o si lo desea, subir una imagen desde su computadora (en la parte de abajo).

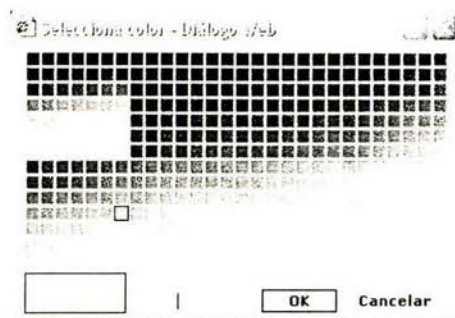


Botón	Función	Descripción
	Propiedades de la imagen	Si incluye una imagen en su mensaje, puede editar propiedades como tamaño, alineación (izquierda, derecha, centrado), ponerle un marco (coloque 1, 2 o 3 en el campo Borde), etc.



	Lista (números)	Genera una lista ordenada en forma numérica
	Lista (balas)	Genera una lista con balas como identificador.
	Sangría (izq.)	Desplaza el texto seleccionado a la derecha.
	Sangría (der.)	Desplaza el texto seleccionado a la izquierda.
	Alineación izquierda	Alinea el texto seleccionado en forma justificada. Es la opción por defecto.
	Alineación centro	Alinea el texto seleccionado de manera centrada.
	Alineación derecha	Alinea el texto a la derecha.

<i>Botón</i>	<i>Función</i>	<i>Descripción</i>
	Color letra	Cambia el color del texto seleccionado.
	Color fondo	Cambia el color del fondo del texto seleccionado. Abrirá una ventana para que seleccione el color de su preferencia.



Limpiador de html	Toda la edición que haga en su texto genera código con formato html (es así como después clientes de correo como Outlook interpretan todo el formato). En ocasiones, se genera código redundante. No es necesario que utilice este botón, de hecho, podría alterar su mensaje.
Cambiar bordes	Si ha insertado una tabla (a continuación verá como hacerlo),

Si requiere utilizar una tabla, el botón para hacerlo se encuentra a la izquierda. Al oprimirlo, se abrirá una ventana en la que debe seleccionar cuántas columnas y filas desea, además del ancho y alto de las celdas (de preferencia, deje estos campos en blanco para que el tamaño sea determinado automáticamente en base al contenido de la celda). Si desea que las celdas tengan borde, coloque 1, 2 o 3 en el campo "Borde de las celdas". También puede poner un color de fondo a la tabla.

Filas: 3 Columnas: 3

Ancho: % Alto: %

Borde: 1 pixels

Borde de las celdas: Espacio entre celdas:

Color de fondo:

OK Cancelar

Si requiere adjuntar algún archivo, debe hacerlo en la siguiente sección.

Cuando haya terminado de escribir el mensaje (incluido asunto y responsable) y haya seleccionado a los destinatarios, oprima “Continuar”.

3.4 Vista previa y envío.

Ahora es momento de confirmar que el mensaje es correcto y los destinatarios también. En esta página verá el asunto del mensaje seguido del tipo de personal al que llegará. A la derecha, puede observar primero a qué subdirección y luego a qué coordinación.

Si desea adjuntar un archivo, en la parte inferior está el botón Adjuntar... Oprímalo para que un navegador se abra y le permita seleccionar el archivo deseado (si sabe la ruta del archivo, bastará con escribirla).

El mensaje se muestra tal como lo verán las personas a las que les sea enviado. Si requiere hacer alguna modificación, en la parte inferior se encuentra el botón “Modificar”, que lo regresará a la página anterior (de nuevo tendrá que elegir el tipo de personal). Si todo es correcto, oprima el botón enviar.

Es muy importante que **no cierre el navegador** hasta que vea un letrero indicándole que el mensaje ya fue enviado (esto puede tomar algunos minutos dependiendo de la cantidad de mensajes enviados).

Referencia Bibliográfica

TEXTOS.

Armstrong, James. UNIX a fondo, Anaya Multimedia, Madrid, 2000

Colado Rodríguez. Diseño y desarrollo de aplicaciones web multidispositivo, Germinus XXI, 2003

Guzmán Castro, Luis Alejandro. Metodología para proyectos de software orientado a objetos basada en CMM y el proceso unificado de desarrollo de software, Tesis de Licenciatura. FI-UNAM, 2003

Graham, Paul. A plan for spam, agosto, 2002

Jacobson, I. Applying UML in The Unified Process, 1998

Letelier, Patricio. Metodologías ágiles y XP, Universidad Politécnica de Valencia

DuBois, Paul. Edición Especial MySQL, Prentice Hall, España, 2001

ROSEN, Kenneth. UNIX Sistema V Versión 4, ed. Mc Graw Hill, México, 1997

SAETHER, Stig. Manual de PHP, Grupo de Documentación de PHP, 2001 (Bajo GNU-GPL)

PÁGINAS.

Aceituno, Vicente. Seguridad de la información, <http://seguridaddelainformacion.net>
consulta: julio de 2004

Brown, Martin. IIS vs. Apache. Looking Beyond the Rhetoric,
http://www.serverwatch.com/tutorials/article.php/10825_3074841_2
2003

Casado Martínez, Carlos. http://www.uoc.edu/mosaic/old/2_opinion2/modeljuny/carlos_casado.html
2004
consulta: julio de 2004

Castellanos Leyva, Alina. <http://www.linux.cu/manual/avanzado-html/node29.html> 2002
consulta: junio de 2004

Chamorro Villar, Ricardo. Análisis comparativo entre Microsoft .NET y SUN J2EE
<http://www.ciberteca.net/articulos/programacion/net/default.asp>
consulta: junio de 2004

Massimiliano Valente. Java y Javascript, http://www.htmlpoint.com/guida/problemi/html_06.htm
2001
consulta: julio de 2004

Teruel, Alejandro. Orientando sistemas de programas a objetos: Un segundo experimento
<http://www ldc.usb.ve/~teruel/ci3711/experimentoLarman.html> Universidad Simón Bolívar, 200
consulta: junio, 2004

Apache License version 2.0, <http://www.apache.org/licenses/#distributions>
consulta: septiembre de 2004

Cross-Browser Rich Text Editor, <http://www.kevinroth.com/rte/demo.php>
consulta: diciembre, 2003

Curso Ingeniería del software, <http://www.eduangi.com/ingsw/tema2/2-4.htm> 2003
consulta: junio de 2004

Database Benchmarking, <http://www.eweek.com/slideshow/0,2394,pg=0&s=700&a=23120,00.asp>
Ziff Davis Media 2004
consulta: agosto de 2004

Desarrolloweb_jsp vs asp, <http://www.desarrolloweb.com/articulos/832.php>
consulta: julio, 2004

Extreme Roles, <http://c2.com/cgi/wiki?ExtremeRoles> 2004
consulta: julio de 2004

Formularios, <http://www.ignside.net/man/html/formularios.php> Ignside, 2004
consulta: marzo, 2004

GerWin, <http://www.es.gnu.org/~jemarch/cgi-bin/index.pl?Site=GerWin> Open TRS Team, 2001
consulta: julio, 2004

Getdate, <http://mx2.php.net/getdate> PHP Group, 2004
consulta: mayo, 2004

Google, <http://www.google.com>
consulta: permanente

Información general SQL Server, <http://www.microsoft.com/latam/sql/evaluation/overview/>
Microsoft Corp. 2003
consulta: julio de 2004

Ingeniería de Software,
<http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html#ProcUnificado>
2000

GNU Public License (GPL), versión 2
junio, 1991
Trad. González Barahona, las Heras Quirós

Misión crítica, http://h30095.www3.hp.com/servicios/servicios_tecnologia/cri_serv_index.html
Hewlett Packard, 2004
mayo, 2004

NTOP, <http://www.ntop.org/ntop.html> Luca Deri, 2002
consulta: junio, 2004

Pear – PHP Extension and application repository, <http://pear.php.net/> 2003
consulta: febrero de 2004

Perl/PHP, getdate, http://perl.about.com/library/phpCR/bl_getdate.htm Primedia Company
consulta: mayo, 2004

PHPMAILER, <http://phpmailer.sourceforge.net/> 2004
consulta: marzo de 2004

PHP MySQL Interactive Website Design, <http://www.createafreewebsite.net/phpmysql/insert.html>
Net Success 2000 plus, 2004
consulta: mayo, 2004

¿Qué se puede hacer con PHP?, <http://www.php.net/manual/es/intro-whatcando.php> The PHP Group,
2004
consulta: junio de 2004

Resumen de metodología Yourdon,
http://trevinca.ei.uvigo.es/~alma/docencia/das/Resumen_Yourdon_Practica.pdf
consulta: julio de 2004

SPAW, http://www.solmetra.com/en/disp.php/en_products/en_spaw/en_spaw_demo Solmetra, 2003
consulta: diciembre 2003

The PHP License, version 3.0, http://www.php.net/license/3_0.txt
1999 - 2004 The PHP Group
consulta: septiembre de 2004

Web server tutorial, <http://www.freeos.com/articles/4121/> Vans information, 2001
consulta: junio de 2004

Wikipedia, la enciclopedia libre, <http://es.wikipedia.org/wiki/API>
consulta: junio de 2004

<http://desarrollo.adesis.com/> Adesis Netlife, 2003
consulta: julio de 2004

<http://dev.mysql.com/tech-resources/benchmarks/> 2004
consulta: junio de 2004

<http://www.gnu.org/software/mailman/mailman.html> Free Software Foundation, 2004
consulta: junio de 2004

<http://www.greatcircle.com/majordomo/> Great Circle Associates Inc, 2003

consulta: mayo de 2004

<http://www.learnthenet.com/spanish/glossary/mime.htm> Michael Lerner Productions, 2004
consulta: mayo de 2004

<http://www.linuxhelp.net/guides/lamp/> Joey Olson, 2004
consulta: junio de 2004

<http://www.macromedia.com/software/homesite/> Macromedia Inc. 2004
consulta: julio de 2004

www.macromedia.com Macromedia Inc, 2004
consulta: julio de 2004

<http://www.microsoft.com/spain/Office/frontpage> Microsoft Corp. 2004
consulta: julio de 2004

http://www.phpmyadmin.net/home_page/ phpMyAdmin devel team, 2003
consulta: julio de 2004

<http://www.rediris.es/list/basico.es.html#mod> RedIRIS, 2004
consulta: mayo de 2004

<http://www.tuwebhost.com/soporte/herramientas-desarrollo-web.htm> Tu Web Host, 2003
consulta: julio de 2004

<http://www.ultraedit.com/> IDM Computer Solutions Inc, 2004
consulta: julio de 2004

KDE HelpCenter 3.00.00
2004

Índice de tablas

Tabla 1.	Organización interna del Instituto de Ingeniería	7
Tabla 2.	Integrantes del Área de Servidores UNIX	11
Tabla 3.	Equipo del Área de Servidores UNIX	11
Tabla 4.	División de personal en base a categoría	13
Tabla 5.	Ventajas y desventajas de mayordomo	19
Tabla 6.	Roles en la programación extrema	27
Tabla 7.	Diferencias entre editores gráficos y de texto	31
Tabla 8.	Editores de texto, híbridos y wysiwyg	32
Tabla 9.	Plataformas web	33
Tabla 10.	Comparación: .net, LAMP, Java	42
Tabla 11.	Comparación: PHP, ASP, JSP	43
Tabla 12.	Comparación sintetizada. IIS, Apache	43
Tabla 13.	Ventajas y desventajas de parejas para autenticación	49
Tabla 14.	Módulos del sistema	55
Tabla 15.	Información para la base de datos	51
Tabla 16.	Tablas previas de la base de datos	60
Tabla 17.	Tabla de verdad, usuario AND contraseña -> salida	68
Tabla 18.	Paso de variables	70
Tabla 19.	Distribución de personal para envío de mensajes	76

Índice de ilustraciones

Figura 1.	Tiempo de respuesta de DBA's respecto al número de usuarios	44
Figura 2.	Páginas (con consulta a base de datos) por segundo respecto a número de usuarios	44
Figura 3.	Banner del sistema	56
Figura 4.	Página de bienvenida	56
Figura 5.	Selección de banner y fondo	57
Figura 6.	La ventana principal de la aplicación	57
Figura 7.	Vista preliminar	58
Figura 8.	Mensaje enviado	58
Figura 9.	Diagrama entidad relación	60
Figura 10.	Página de inicio phpMyAdmin	61
Figura 11.	Creación de la tabla personal	62
Figura 12.	Ventana de autenticación	64
Figura 13.	Barra de botones SPAW	65