

Metodología para desarrollar e implementar un servicio de correo electrónico en organizaciones

A methodology to develop and implement an email service

Recibido: 11/07/2016

Aceptado: 15/12/2016

Resumen: Se presenta una metodología para implementar un servicio de correo electrónico con mínimos recursos computacionales. La utilidad de este servicio consiste en que está pensado para que sea utilizado como correo interno de cualquier organización, que pudiera expandirse, con algunas modificaciones, a un servicio que pudiera utilizar cualquier persona. La metodología descrita surgió de la necesidad del Instituto de Estadística Aplicada y Computación, IEAC, adscrito a la Facultad de Ciencias Económicas y Sociales de la Universidad de Los Andes, de Mérida, Venezuela, de ahorro del uso del papel y de tener a disposición un mecanismo electrónico en la world wide web, por medio del cual su personal intercambiara documentos contentivos de información de sus actividades diarias. Esta necesidad se tradujo en diseñar y construir un conjunto de programas que dio lugar a este servicio de correo para el IEAC. Durante la búsqueda de cómo llevar a cabo esta aplicación no se encontró una forma precisa, académica y sin rodeos de los procedimientos involucrados en este tipo de trabajos, razón por la cual, con la experiencia adquirida y a partir del producto obtenido, se propone esta metodología como una contribución al desarrollo de estos servicios. En la actualidad, esta aplicación está operativa en el IEAC.

Palabras clave: Metodología, correo-electrónico, interfaces, usuario, archivos

Abstract: This paper presents a methodology that describes how to implement an email service with minimal computational resources. The usefulness of this service resides in the fact that it is intended to be used as an internal email of any organization, although it could also be expanded, with some modifications, to a service available to anyone. The methodology described emerged from the need that the Instituto de Estadística Aplicada y Computación, IEAC, attached to the Facultad de Ciencias Económicas y Sociales de La Universidad de Los Andes, Mérida, Venezuela, had, in saving the use of paper and to make available a world wide web based electronic mechanism, through which its staff could exchange documents of their daily activities. These needs led to design and build a software that resulted in this email service for IEAC. When searching for how to carry out this application it was not found an accurate, academic and bluntly procedure involved with this type of work, reason why, with the experience gained and from the product obtained, this methodology is proposed as a contribution to the development of these kind of services. Currently, this application is being used in the IEAC.

Keywords: Methodology, e-mail, interfaces, user, files

Núñez, Yesenia y Maldonado, Luciano
Universidad de Los Andes (ULA), Instituto de Estadística Aplicada y
Computación (IEAC), FACES, Mérida - Venezuela
yesenian@ula.ve / maldonaj@ula.ve

I. INTRODUCCIÓN

En el mundo tecnológico actual, el mundo de la internet, muy pocas personas escapan al uso electrónico de la información. Por esta razón, la mayoría de las empresas, y en general, toda institución pública o privada requiere de medios que coloquen a disposición, de manera inmediata y segura, datos que cualquier miembro del personal necesite en un momento determinado. En este sentido, es bien conocido que las redes sociales han venido jugando un rol determinante en los últimos años por estar al alcance, prácticamente, de cualquier persona, pero con la desventaja de que, precisamente, por ser un medio de uso absolutamente masivo, se ha convertido en un factor de distracción y que puede influenciar, también, negativamente el desarrollo de las actividades propias de las instituciones. Por esta razón, el servicio de correo electrónico recobra su importancia a la hora de intercambiar información, en forma digital, lejos de la informalidad con las que se utilizan las redes sociales (Brito y Gamess, 2016, Boyce, 2003, Lee, 2014, Silvestri, Guzman, Piedrabuena y Di Biase, 2014).

Por otro lado, aun cuando existen grandes empresas especializadas en el desarrollo de servicios de correo electrónico, incluso gratuitos, hay una realidad que muy pocas veces se toma en cuenta, como es el servicio de correo interno o particular para una empresa, institución o simplemente para un grupo cualquiera de trabajo o de estudio. Ese tipo de servicio de correo está orientado a un determinado conjunto de personas, cuyos integrantes sólo pueden usarlo a través de alguna autorización, por ejemplo, de algún directivo de la institución o de un coordinador de un grupo de trabajo o de estudio. Como valor agregado, estando este tipo de servicio dirigido a un grupo cualquiera de trabajo o de estudio, jugaría un rol de primerísima línea como herramienta para complementar aquellas que actualmente se vienen utilizando como medios para impartir educación, tanto a distancia como presencial, puesto que se constituye en un medio directo, y controlado, para la comunicación electrónica entre alumnos, y entre profesores y alumnos.

En este trabajo se presenta una metodología orientada al desarrollo de aplicaciones web que permite implementar el servicio de correo interno, o específico, para alguna institución, con un enfoque muy distinto a los servicios de correo tradicionales, puesto que no se recurre a las herramientas tecnológicas, ampliamente conocidas, asociadas a esos servicios (Brito, 2016, Boyce, 2003, Lee, 2014, Silvestri *et al.*, 2014). En este sentido, se trata de un enfoque particular y sencillo de desarrollo de estos servicios con un doble propósito: que el personal de la institución se enfoque en su actuación netamente institucional y evitar retardos indeseables en las respuestas, situación que se puede presentar cuando se utilizan los servicios masivos puesto que, en la mayoría de los casos, para tratar información local de una institución igual hay que recurrir a servidores externos, aun cuando se cuente con servidores locales.

Para explicar la metodología propuesta se describe el servicio de correo implementado en el Instituto de Estadística Aplicada y Computación, IEAC, adscrito a la Facultad de Ciencias Económicas y Sociales, de la Universidad de Los Andes de Mérida, Venezuela. Este Instituto es el encargado de dictar, en esta Universidad, los Programas de Maestría y Doctorado en Estadística.

II. DESCRIPCIÓN DE LA METODOLOGÍA

Como ocurre con cualquier metodología, la metodología propuesta para desarrollar e implementar un

servicio de correo electrónico, como el planteado, sugiere una serie de pasos.

Es importante resaltar que esta metodología surge como una alternativa al tradicional desarrollo de servicios de correo, para el cual en la *www* se dispone de amplia información, en cuanto al procedimiento, para realizar esta tarea (Brito, 2016, Boyce, 2003, Lee, 2014, Silvestri *et al.*, 2014).

Etapas de la metodología

A continuación se describen las etapas que conforman la metodología:

A. Solicitar espacio en un servidor web donde alojar el servicio de correo

Como toda aplicación web, el servicio de correo necesita almacenar su información en un servidor web y de bases de datos. Por lo tanto, la primera actividad que se debe cumplir es la de solicitar un espacio en un servidor institucional o público, gratuito o no. En el caso del servicio de correo desarrollado para el IEAC, se solicitó espacio en un servidor de la Universidad de Los Andes, de Mérida, Venezuela; pero pudo haberse contratado, por ejemplo, con la estatal venezolana CANTV o haberse recurrido a cualquiera de los servidores que prestan este servicio de manera gratuita, alrededor del mundo, corriendo el riesgo de lo que esto último implica.

B. Selección de los recursos de software para el desarrollo

Para este tipo de desarrollos se puede recurrir a recursos de software libre o comercial, según la preferencia, de última generación o simplemente aquel que se tenga a disposición. En cualquier caso, hay que contar con al menos un editor de texto, un manejador de bases de datos, un compilador o interpretador de algún lenguaje de programación, software para transferencia de archivos, lenguajes de estilos y de etiquetas: los recursos tanto del lado del cliente como del lado del servidor.

En la implementación del servicio de correo, a través del cual se describe la metodología, se recurrió a *mysql*, *php*, *html*, *css*, *javascript*, *pluggins*, *jquery*, *notepad++*, *las librerías fpdf* y *FileZilla*. En todo caso, es el desarrollador quien decide qué recursos de software utilizar.

C. Crear una base de datos de usuarios

Antes de crear un servicio de correo electrónico, se tiene que definir a qué usuarios está orientado ese servicio. En el trabajo que se está describiendo, el correo fue creado para ser utilizado por las personas adscritas al IEAC solamente.

Para el grupo de usuarios, como el del IEAC, hay que crear una base de datos que contenga la información relevante de esos usuarios, tal como nombres, cédula de identidad y cargo.

La construcción de esa base de datos de usuarios, conlleva la implementación de un módulo que permita inscribir al usuario, mostrar su registro, modificar los datos registrados de un usuario y eliminar el registro del usuario. Para ello se requiere de un manejador de bases de datos, por ejemplo, *mysql*, como fue en el caso del correo del IEAC. Para construir este módulo se requiere, también, de una interface de usuario que le permite al administrador del correo, realizar las actividades mencionadas.

D. Crear las cuentas de los usuarios del correo

Para crear las cuentas de los usuarios existen varias alternativas, entre éstas, por ejemplo, la forma en que se crean para el correo del IEAC. Para este caso, cada usuario registrado en la base de datos puede crear su propia cuenta proporcionando el nombre y apellido, la cédula o pasaporte y su clave de usuario. Mientras que otra alternativa es disponer de una persona que administre las cuentas de los correos. En cualquier caso, hay que construir una interface que permita crear la cuenta, eliminar la cuenta, modificar los datos de la cuenta y mostrar los datos de la cuenta. Para ello se puede recurrir, por ejemplo, al uso de *html*, *css*, *notepad++* y *php* para el desarrollo de la interface propiamente dicha; y *mysql* para las consultas, inserciones, modificaciones y eliminaciones de los registros en la base de datos de las cuentas de los usuarios del correo. En la Fig. 1 se muestra la interface a través de la cual un usuario crea su cuenta en el correo del IEAC.

Una vez que se ejecuta la inserción de los datos que ingresa el usuario, en la base de datos, para crear su cuenta de correo se crea, a través de programación, la carpeta del usuario en el servidor en la que se almacenarán sus mensajes, tanto recibidos como enviados. Una forma de hacer esto puede ser, por ejemplo, a través de la función *mkdir* en *php*, especificando la ruta correspondiente y definiendo los permisos de accesos respectivos para esas carpetas (Subir múltiples archivos con php, 2016).



Crear Cuenta IEAC

Nombre y Apellido:

Cédula/Pasaporte:

Clave de correo:

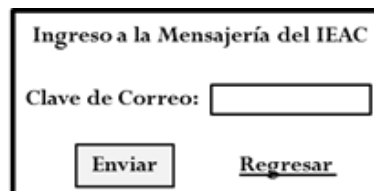
[Regresar](#)

Figura 1. Interface para crear la cuenta de correo

E. Construcción de la Interface para uso del correo

Para que un usuario pueda hacer uso del correo hay que desarrollar una interface que le permita acceder, a través de su clave, a los servicios de éste. Para ello, también, se puede recurrir al uso de *html*, *css*, *javascripts* y *jquery* para el desarrollo de la interface; conjuntamente con *php* y *mysql* para la consulta y verificación de la clave del usuario.

En la Fig. 2 se muestra la interface para ingresar al correo del IEAC.



Ingreso a la Mensajería del IEAC

Clave de Correo:

[Regresar](#)

Figura 2. Interface de Inicio de Sesión

F. Implementación de los servicios de correo

El conjunto típico de servicios de un correo puede resumirse por medio de la descripción del correo desarrollado para el IEAC. Este conjunto consiste en las tareas de redacción de mensajes, lectura de mensajes nuevos, lectura de mensajes ya leídos, eliminación de mensajes y lectura de mensajes enviados. Además, tiene que incluirse el tratamiento de los archivos adjuntos. Estos servicios se ofrecen a los usuarios a través de una interface basada, por ejemplo, en un menú de opciones implementado en los módulos que se explican a continuación:

- Módulo para la redacción de mensajes.

El diseño e implementación de un módulo para la redacción de mensajes tiene como componente principal una interface de usuario. Esta interface consta de tres campos obligatorios: el destinatario, un área de texto para escribir el mensaje, y una opción para adjuntar archivos al mensaje. Estos componentes se muestran en la Fig. 3.

The image shows a web form for composing an email. It consists of several labeled input fields and buttons. At the top, there is a 'Destinatario:' label followed by a single-line text input field. Below that is a 'Mensaje:' label followed by a larger multi-line text area. Underneath the message area is an 'Adjuntos:' label, followed by two buttons: 'Seleccionar Archivo' and 'Añadir'. At the bottom of the form are two buttons: 'Enviar' and 'Regresar'.

Figura 3. Interface para redactar mensajes

Tomando esta estructura como referencia, el procedimiento para la construcción de este módulo se explica a continuación.

1. El destinatario, en un correo electrónico, se identifica por la cuenta del usuario al que se le va a enviar el mensaje. Un procedimiento sencillo aplicado en el Correo del IEAC, consiste en usar como cuentas los nombres y apellidos de las personas que han sido registradas en la base de datos como usuarios del correo.

Para seleccionar el destinatario, básicamente, el procedimiento consiste en, a partir de una consulta a la base de datos, mostrar los usuarios, registrados, a través de, por ejemplo, un *Select* Dinámico múltiple de *html*, en una lista desplegable. Una vez que se obtiene la lista desplegable, se selecciona uno o varios destinatarios.

También, se pueden fijar grupos de destinatarios para correos institucionales como, por ejemplo, en el caso del IEAC en el cual se agrupan destinatarios por categorías de profesores, personal administrativo, técnico y obrero. En este caso, la diferencia con la selección de destinatario, previamente explicada, consiste en que hay que asociar cada miembro de un grupo con un índice en el momento en que se registra el usuario en la base de datos. Posteriormente, ese índice se

emplea para determinar, a través de una consulta a la base de datos, los destinatarios miembros del grupo al que se le enviará un mensaje. Una vez determinados los miembros del grupo de destinatarios, se le envía a cada uno de esos miembros el mensaje de interés común.

Para la consulta de los usuarios se puede utilizar *mysql*, para seleccionar los destinatarios puede utilizarse *html* y para mostrar, de manera dinámica, la lista desplegable se puede usar *php*.

2. Para diseñar el área de texto del mensaje se puede recurrir al uso del *TextArea* de *html* para escribir y editar las líneas del mensaje a enviar (*Áreas de texto*, 2016).

3. Finalmente, el procedimiento para adjuntar archivos a un mensaje tiene que tomar en cuenta si se va a enviar uno o varios archivos a un destinatario, a varios destinatarios o a un grupo definido de destinatarios.

Para adjuntar los archivos asociados a un mensaje se encuentran diversas herramientas disponibles en la *www*. En el caso de la aplicación que se describe se utilizó el código *javascript* encontrado en la *url* siguiente: <http://www.cristalab.com/tutoriales/subir-multiples-archivos-con-php-c323811/> (Subir múltiples archivos con *php*, 2016).

El mecanismo a seguir para enviar un mensaje y un adjunto a un destinatario, en el correo del IEAC, es como sigue: una vez que se tiene a disposición el mensaje, el destinatario seleccionado y el archivo adjunto, se envía el mensaje insertando en la base de datos un registro correspondiente al destinatario. Este registro contiene el mensaje, el nombre del remitente, la fecha y hora del servidor, el número del mensaje que el remitente le envía al destinatario (este número se obtiene consultando cuántos mensajes le ha enviado un remitente a un destinatario) y el nombre del archivo adjunto al mensaje. También, se traslada el archivo adjunto, físicamente, a la carpeta del usuario, destinada para tal fin, que se encuentra en el servidor.

Cuando el mensaje no contiene archivos adjuntos se sigue el mismo procedimiento, con la diferencia que no se registra nombre de archivo adjunto ni se traslada ningún archivo físicamente a la carpeta del usuario.

Es importante identificar cada mensaje recibido, por un destinatario, con una condición que señale el estado de ese mensaje como *no leído* o como *ya leído*. Los no leídos serían aquellos mensajes no consultados por el usuario, mientras que después de consultados, la primera vez, se cambian al estado de leídos a través de la instrucción *update* del manejador de bases de datos. Por otro lado, por todo mensaje enviado, por un remitente, se crea en la base de datos, una copia con una condición que señala que se trata de un mensaje enviado por ese remitente. Esto se hace para que el remitente pueda identificar sus mensajes enviados.

Este procedimiento es el mismo para enviar un mensaje con varios adjuntos a varios destinatarios o a un grupo de destinatarios, con la diferencia de que previamente hay que realizar, primero una selección dinámica y múltiple de los destinatarios, y posteriormente, se van seleccionando uno a uno los archivos a enviar. Luego, por cada uno de los destinatarios, hay que insertar en la base de datos de los usuarios el mensaje y una cadena que contiene, en forma concatenada, los nombres

de los respectivos archivos adjuntos. Además, se traslada cada archivo adjunto, físicamente, a la carpeta del usuario en el servidor.

Una consideración a tomar en cuenta, en este tipo de desarrollos, es que se debe determinar cuál es el tamaño máximo de los archivos adjuntos, dependiendo de la disponibilidad de espacio en el servidor, e informar a los usuarios a través de alguna leyenda en la interface.

- Módulo para lectura de mensajes nuevos

Este servicio pone a disposición del usuario la información de sus mensajes recibidos, y que todavía no han sido leídos. Una forma de mostrar estos mensajes es a través de una interface o en un documento pdf como se hace en el correo del IEAC.

Para construir ese archivo pdf se puede recurrir a cualquier librería disponible para tal fin, como es el caso de la librería FPDF, disponible en <http://www.fpdf.org/> (librería FPDF, 2016), para la cual se describe, a continuación, la forma de utilizarla conjuntamente con *php* para mostrar los mensajes nuevos:

1. Se descarga a una de las carpetas asociadas al conjunto de programas de la aplicación del correo que se esté desarrollando.
2. Se incluye la ruta de la librería *fpdf* en el programa desarrollado para construir el documento pdf.
3. Se crea el objeto de la clase *FPDF* como `$pdf=new FPDF()`.
4. Cada página del documento se inicializa con `$pdf->AddPage()`.
5. Se establece el formato de cada página del documento en el que aparecerán los datos de los mensajes.
6. Se consulta en la base de datos los mensajes no leídos del usuario.
7. Se escriben los mensajes no leídos del usuario en el documento pdf, utilizando los métodos de la clase pdf de la librería *fpdf* que se basa en la función `$pdf->Cell()`, estableciendo el ancho y el alto de la celda donde aparecerán los datos.
8. En el caso de existir archivos adjuntos se recurre al Módulo para la Lectura de Archivos Adjuntos, que se describe posteriormente a este apartado, a través del cual se genera la lista de los archivos que se van a descargar.
9. Por último, se recurre a la función `$pdf->Output()` para generar el archivo pdf y enviarlo al navegador. En el documento pdf, al pulsar sobre los links, quedan a disposición los archivos adjuntos.

- Módulo para la lectura de archivos adjuntos

Para el caso de aquellos mensajes que contienen archivos adjuntos, se puede usar la siguiente estrategia para la lectura de estos archivos:

1. Consultar en la base de datos el campo del registro, asociado al mensaje a leer y al usuario, que contiene la cadena de caracteres con los nombres de los archivos adjuntos.
2. Separar de la cadena de caracteres cada uno de los nombres de los archivos adjuntos a recuperar. Para ello se puede recurrir, por ejemplo, al comando *explode* de *php*.
3. Por cada uno de los archivos adjuntos se localiza, en el servidor, este archivo en la carpeta asociada al usuario que está leyendo el mensaje. Para ello, previamente, se construye la ruta de la carpeta específica para el usuario.

A continuación se muestra un ejemplo que describe esta actividad, trabajando con *php*, *mysql* y las librerías *fpdf*:

```
$adjunto="http://www.faces.ula.ve/ieac/mensajeria/usuarios/" . $usuario_actual . '/' . $arreglo_adjuntos[$j]
```

En el ejemplo, se puede observar que

http://www.faces.ula.ve/ieac/mensajeria/usuarios/, constituye la ruta a una carpeta que contiene las carpetas de todos los usuarios. *\$usuario_actual* . '/' , constituye la ruta a la carpeta del usuario, específico, que está leyendo el mensaje en un momento dado y *\$arreglo_adjuntos*[*\$j*], corresponde al nombre del archivo adjunto *j* a descargar.

Esta información, obtenida para cada uno de los archivos adjuntos, se deja disponible en un arreglo para ser utilizado por el Módulo para lectura de mensajes nuevos encargado de mostrar, finalmente, los mensajes nuevos y dejar disponibles los archivos adjuntos para su descarga. Se debe acotar que se mostrarán tantos links de descarga como archivos adjuntos existan.

- Módulo para lectura de mensajes ya leídos

Este servicio pone a disposición del usuario la lista de sus mensajes ya leídos, que han sido almacenados con la finalidad de tenerlos a disposición para cualquier consulta posterior. Para construir esa lista se sigue el mismo procedimiento descrito para el caso de la lectura de mensajes nuevos, con la diferencia de que en esta oportunidad la consulta a la base de datos se realiza sobre mensajes que cumplen con la condición de ya haber sido leídos.

La Fig. 4 muestra la lista de mensajes leídos de un usuario y sus adjuntos, a través de un documento pdf.

LISTA DE MENSAJES LEÍDOS				
REMITENTE	Nº. MENSAJE	MENSAJE	FECHA	HORA
Luciano Maldonado	1	Yesenia, Por favor pruebe el módulo del correo del Proyecto IEAC -Electrónico. Luciano	2016-05-17	05:33:00
Lucina Flores	1	Buenas tardes, le adjunto archivo. Saludos	2016-04-30	02:15:00
ADJUNTOS: http://www.faces.ula.ve/ieac/mensajeria/usuarios/yesenia_nunez/articulo_correo.docx				

Figura 4. Documento pdf con mensajes leídos

- Módulo para lectura de mensajes enviados

Este servicio pone a disposición del usuario sus mensajes enviados a otros usuarios o a sí mismo. Para construir esa lista de mensajes se sigue el mismo procedimiento descrito tanto para el caso de la lectura de mensajes nuevos como para los ya leídos, con la diferencia de que la consulta a la base de datos se realiza sobre mensajes que cumplen con la condición de enviados sólo por el usuario que esté realizando la consulta. Esta consulta debe realizarse ordenando los registros, por ejemplo, por destinatario y por fecha de envío, mostrando primero los más recientes.

- Módulo para la eliminación de mensajes

Este módulo se puede implementar a través del uso de varias interfaces en cascada:

1. Una interface que permita mostrar a través de un menú de opciones, los tipos de mensajes a eliminar, por ejemplo, eliminar los mensajes enviados o eliminar los mensajes leídos.
2. Una interface que permita seleccionar, dependiendo si se trata de mensajes leídos o de mensajes enviados: un mensaje de un usuario específico o todos los mensajes, enviados o leídos, del usuario que está haciendo la consulta.
3. Una interface para eliminar un solo mensaje leído o enviado: Para eliminar un mensaje leído, se selecciona el remitente y se especifica el mensaje a eliminar. Para eliminar un mensaje enviado, se selecciona el destinatario y se especifica el mensaje a eliminar.
4. Una interface para eliminar todos los mensajes recibidos por el usuario o todos los mensajes enviados por el usuario.

En todos los casos es conveniente colocar un mensaje de advertencia antes de dar la orden de eliminar.

III. FORMAS DE MANTENER LA IDENTIDAD DEL USUARIO ENTRE LAS DISTINTAS PÁGINAS QUE CONFORMAN LA APLICACIÓN DESARROLLADA BAJO ESTA METODOLOGÍA

Como es bien conocido, en una aplicación web, al pasar de una página a otra los datos recibidos a través de un formulario, por medio de los comandos, por ejemplo, GET y POST de *html* y *php*, se pueden mantener para cualquier tratamiento sobre ellos en la página que los recibe. Sin embargo, hay datos que por sus características se requiere que adquieran la propiedad de ser globales, es decir, que se mantengan para todas las páginas y scripts que conforman la aplicación. Para lograr esta propiedad existen diferentes mecanismos, entre los cuales se encuentra el uso de campos ocultos en formularios, como se muestra en el ejemplo siguiente:

```
<form method="post" action="script.php?usuario=<?php echo $identidad;?>"
```

```
...
```

```
Cuerpo del formulario
```

```
...
```

```
</form>
```

En este ejemplo, en un formulario cualquiera y en un script.php cualquiera, el contenido de la variable *\$identidad* se pasa a este script a través de la variable usuario.

Por otro lado, el script.php realiza la recepción de la siguiente manera:

```
$identidad=$_GET["usuario"]
```

En el ejemplo descrito se presenta el mecanismo para mantener la identidad de un usuario, entre dos páginas de una aplicación, pero puede lograrse esta capacidad para múltiples páginas de una aplicación siguiendo el mismo procedimiento (Manteniendo el estado de las solicitudes a un servidor web, 2016).

Al seguir la estrategia descrita, la aplicación adquiere la capacidad de simultaneidad de un Servicio de Correo Multiusuario donde, al momento de consultas a la bases de datos, cada usuario puede obtener exactamente los datos que sus solicitudes de lectura de mensajes requieren. Y lo mismo sucede en el caso de las inserciones para los envíos de mensajes.

Para el caso del correo del IEAC se trabajó en un principio con sesiones, obteniendo malos resultados, por lo que se recurrió a la técnica de los campos ocultos en formularios.

IV. LINKS A TRAVÉS DE LOS CUALES SE PUEDE ACCEDER AL SISTEMA QUE SIRVIÓ DE EJEMPLO PARA EXPLIACR LA METODOLOGÍA DESCRITA

La página web del Instituto de Estadística Aplicada, IEAC, de la Universidad de Los Andes, de Mérida, Venezuela, se encuentra en la siguiente URL:

<http://www.faces.ula.ve/ieac/>

En el menú de opciones de esta página se encuentra la opción correspondiente al acceso al correo interno de este Instituto.

V. CONCLUSIONES

La metodología propuesta se convierte en un recurso que viene a llenar un vacío, en la forma a seguir para construir los servicios de correo interno de organizaciones particulares, en tanto que evita, en algún grado, la dependencia de los protocolos y el control de la información que caracterizan a los servicios tradicionales.

Esta metodología puede ser expandida para desarrollar un servicio de correo más general, sin grandes modificaciones con un enfoque totalmente distinto al empleado en la implementación de los tradicionales servicios de correos.

El servicio de correo elaborado para el IEAC, que sirve de referencia para explicar la metodología, se encuentra operativo, y tiene una velocidad de respuesta superior a cuando se utilizan los servicios de correos tradicionales. Por lo tanto, se hace evidente que este enfoque de trabajo puede servir de punto de partida para que cualquier organización, con poca disponibilidad de recursos computacionales, elabore su propio servicio de correo interno.

El diseño y construcción de este tipo de aplicación requiere de suficiente conocimiento del área de los desarrollos web, que permita entender y programar las necesidades multiusuario tanto del lado del cliente como del lado del servidor.

El servicio de correo propuesto saca ventaja de la parte de la tecnología web orientada exclusivamente a sistemas de información, cuyo funcionamiento depende de manejadores de bases de datos y esto, es precisamente, lo que hace diferente este enfoque a los tradicionales servicios de correos. Es decir, el tipo de servicio de correo que se propone no hace uso de los protocolos existentes, propios de los servicios de correos, sino que se construyen todos los programas recurriendo a las tecnologías asociadas al tipo de desarrollo de aplicaciones web donde se requiere de manejadores de bases de datos.

VI. REFERENCIAS

Áreas de Texto. Disponible en línea: <http://www.webestilo.com/html/cap7d.phtml>. Consultada en mayo de 2016.

Boyce J. (2003). *Get IT Done: Host your own e-mail server*. Disponible en línea: <http://www.techrepublic.com/article/get-it-done-host-your-own-e-mail-server/>. Consultada en febrero de 2016.

Brito A. y Gamess E. *Implementación de un sistema para el manejo de correo electrónico con autenticación centralizada basada en servicios de directorio*. Disponible en línea: <http://cicore.ciens.ucv.ve/wp-content/uploads/2013/05/TEGs/TEG-AlejandroBrito.pdf>. Consultada en febrero de 2016.

Manteniendo el estado de las solicitudes a un servidor web. Disponible en línea: http://programacion-php.readthedocs.io/es/latest/Tutorial3_Sesiones.md.html. Consultada en mayo de 2016.

Lee H. (2014). *How to run your own e-mail server with your own domain*. Disponible en línea: <http://arstechnica.com/information-technology/2014/02/how-to-run-your-own-e-mail-server-with-your-own-domain-part-1/>. Consultada en febrero de 2016.

Librería FPDF. Disponible en línea: <http://www.fpdf.org/>. Consultada en marzo de 2016.

Silvestri L., Guzman S., Piedrabuena M. y Di Biase J. (2014). *SquirrelMail, nuestro propio servidor de mails*. Disponible en línea: <http://www.interorganic.com.ar/josx/SquirrelMail.pdf>. Consultada en febrero de 2016.

Subir múltiples archivos con php. Disponible en línea: <http://www.cristalab.com/tutoriales/subir-multiples-archivos-con-php-c323811/>. Consultada en marzo de 2016.

Como citar este artículo:

Núñez, Y. y Maldonado, L. (2016). Metodología para desarrollar e implementar un servicio de correo electrónico en organizaciones. *Aprendizaje Digital*, Vol 1 N°2, pp 9 – 20.