



**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE CÓMPUTO**

**ESCOM**

*Trabajo Terminal*

**“Sistema editor de presentaciones en línea”**

*Que para cumplir con la opción de titulación curricular en la  
carrera de*

**“Ingeniería en Sistemas Computacionales con Especialidad en  
Electrónica”**

*Presenta*

**Luján Domínguez Víctor Manuel  
Martínez Hernández Yair Elohim  
Ruz Hernández Miguel de Jesús**

*Directores*

**M. en C. Mónica Rivera De la Rosa  
José Marco Antonio Rueda Meléndez**

*México D.F., a 17 de Diciembre del 2010*



**INSTITUTO POLITÉCNICO NACIONAL**  
**ESCUELA SUPERIOR DE CÓMPUTO**



**No. De registro: TT 11-1-0012**

**Diciembre 2010**

Documento técnico

**Sistema editor de presentaciones en línea**

Presenta

**Luján Domínguez Víctor Manuel<sup>1</sup>**  
**Martínez Hernández Yair Elohim<sup>2</sup>**  
**Ruz Hernández Miguel de Jesús<sup>3</sup>**

Directores

**M. en C. Mónica Rivera De la Rosa**  
**José Marco Antonio Rueda Meléndez**

Resumen

El presente Trabajo Terminal propone el desarrollo de una aplicación Web la cual será capaz de realizar presentaciones digitales en línea y fuera de línea. El editor contará con diversas plantillas y estilos prediseñados para un diseño rápido de las presentaciones y diapositivas.

El sistema esta destinado a cualquier usuario con la necesidad de realizar una presentación y no cuente con el software para realizarlo o la compatibilidad para usar dicho software. El usuario tendrá la facilidad de utilizarlo ya sea por medio del Internet o en su escritorio descargando la aplicación. Será capaz de guardar las presentaciones en un formato estándar ODF para que las presentaciones puedan ser editadas por otro software como Open Office. El uso del sistema será de forma gratuita.

Entre los alcances se tienen previstos que el software sea capaz de realizar presentaciones digitales en línea y fuera de línea (aplicación de escritorio), guardar las presentaciones con en el estándar ODF compatible con Open Office y Microsoft Office. Se espera que los usuarios puedan compartir sus presentaciones en línea con los diferentes usuarios.

Palabras Claves: XML, JavaScript, ODF, Programación Web, diapositivas.

---

1 resistencia\_np@hotmail.com

2 capitan\_nice@msn.com

3 ruzsystem\_08@yahoo.com.mx

## ÍNDICE

CAPITULO I. Introducción.....	- 1 -
1.1 Introducción.....	- 1 -
1.2 Justificación.....	- 2 -
1.3 Objetivos.....	- 2 -
1.3.1 Objetivo General.....	- 2 -
1.3.2 Objetivos Específicos.....	- 2 -
1.4 Estado del Arte.....	- 2 -
CAPITULO II. Marco Teórico.....	- 4 -
2.1 Software en Línea.....	- 4 -
2.2 Hiper Text Markup Language (HTML).....	- 5 -
2.3 Open Document Format (ODF).....	- 6 -
2.4 Tecnologías lado servidor.....	- 7 -
2.4.1 PHP Hypertext Pre-processor (PHP).....	- 7 -
2.5 Tecnologías lado cliente.....	- 10 -
2.5.1 JavaScript.....	- 10 -
2.5.2 Cascading Style Sheets (CSS).....	- 11 -
2.5.3 Frameworks.....	- 12 -
2.5.3.1 Características de los frameworks.....	- 13 -
2.5.4 Ashynchronous JavaScript And XML (AJAX).....	- 14 -
2.5.5 Framework Prototype.....	- 15 -
2.5.6 Framework Mootools.....	- 15 -
2.5.7 Framework jQuery.....	- 16 -
2.6 Base de datos.....	- 18 -
2.6.1 Sistema Gestor de Base de Datos.....	- 18 -
2.6.2 MySQL.....	- 18 -
CAPITULO III. Análisis.....	- 20 -
3.1 Metodología.....	- 20 -
3.2 Formulación.....	- 20 -
3.3 Análisis.....	- 20 -
3.3.1 Análisis de contenido.....	- 20 -
3.3.2 Análisis de interacción.....	- 20 -
3.3.3 Análisis funcional.....	- 21 -
3.3.4 Análisis de configuración.....	- 24 -
3.3.4.1 Requerimientos Funcionales.....	- 24 -
3.3.4.2 Requerimientos No funcionales.....	- 24 -
3.3.4.3 Análisis.....	- 25 -
3.3.4.4 Análisis de Factibilidad Técnica.....	- 25 -
3.3.4.5 Análisis de Factibilidad.....	- 26 -
3.3.4.6 Conclusión de factibilidad.....	- 27 -
CAPITULO IV. Diseño.....	- 28 -
4.1 Diseño Arquitectónico.....	- 28 -
4.2 Estructura de una diapositiva.....	- 29 -
4.3 Estructura de un archivo.....	- 30 -
4.4 Generar ODP.....	- 31 -
4.5 Diseño de Interfaz.....	- 33 -
4.5.1 Interfaz LogIn Wediapz.....	- 33 -
4.5.2 Presentaciones de usuario.....	- 34 -
4.5.3 Editor Wediapz.....	- 35 -

4.6 Base de Datos .....	- 37 -
4.7 Estructura general de directorios del sistema .....	- 38 -
4.8 Compartir presentaciones .....	- 40 -
4.9 Recuperar contraseña.....	- 41 -
CAPITULO V. Pruebas .....	- 42 -
5.1 Uso del editor Wediapz.....	- 42 -
5.2 Lectura del formato ODP .....	- 43 -
5.3 Crear nuevas diapositivas .....	- 44 -
5.4 Guardar en formato ODP.....	- 44 -
Conclusiones.....	- 45 -
Trabajo a futuro .....	- 45 -
Glosario .....	- 46 -
Referencias .....	- 48 -

## **Advertencia**

*“Este informe contiene información desarrollada por la Escuela Superior de Cómputo del Instituto Politécnico Nacional a partir de datos y documentos con derecho de propiedad y por lo tanto su uso queda restringido a las aplicaciones que explícitamente se convengan.”*

La aplicación no convenida exime a la escuela su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen.

Información adicional sobre este reporte técnico podrá obtenerse en:

La Subdirección Académica de la Escuela Superior de Cómputo del Instituto Politécnico Nacional, situada en Av. Juan de Dios Bátiz s/n.

## Agradecimientos

*Agradezco profundamente a mis padres Alicia y Juan Manuel por su apoyo directa o indirectamente durante la elaboración de este proyecto, y a lo largo de toda mi trayectoria académica. Sin el apoyo de ellos nunca hubiera podido alcanzar mis objetivos profesionales.*

*De igual forma quiero agradecer a mis compañeros de Trabajo Terminal Yair y Ruz por todo el tiempo y esfuerzo que invirtieron; así como por todo lo que me permitieron aprender de ellos.*

*A mis amigos con los cuales compartí grandes momentos dentro y fuera de la escuela, por ayudarme y estar conmigo a lo largo de la carrera.*

*Luján Domínguez Víctor Manuel*

## Agradecimientos

*A mis padres a quienes jamás encontraré la forma de agradecer el que me hayan brindado su mano en las derrotas haciendo de este triunfo más suyo que mío por la forma en la que han guiado mi vida.*

*... y a todas aquellas personas que comparten conmigo este triunfo.*

*Martínez Hernández Yair Elohim*

## Agradecimientos

*Este Trabajo Terminal, si bien ha requerido de esfuerzo y mucha dedicación por parte de los autores, no hubiese sido posible su finalización sin la cooperación desinteresada de varias personas que han sido un soporte muy fuerte en momentos de angustia y desesperación.*

*Agradezco hoy y siempre a mi familia, novia y amigos, se que procuran mi bienestar, y está claro que si no fuese por el esfuerzo realizado por ellos, mis estudios no hubiesen sido posibles. Y porque a pesar de todo, el ánimo, apoyo y alegría que me brindan me dan la fortaleza necesaria para seguir adelante.*

*Ruz Hernández Miguel de Jesús*





# **CAPITULO I. Introducción**

## **1.1 Introducción**

En la actualidad el poder del Internet ha llegado a niveles inimaginables, los diferentes usos que tiene desde la comunicación a diferentes puntos del mundo, hasta la creación de software en línea que ofrecen un servicio determinado a cualquier usuario que este dentro de esta gran nube.

En este trabajo se desarrollará una aplicación Web, capaz de realizar presentaciones en línea y poder editar también las presentaciones fuera de línea. Este servicio podrá utilizarse de forma totalmente gratuita, siendo una herramienta muy útil ya que con solo un navegador el usuario contará con sus presentaciones sin la preocupación de compatibilidad de otro software o la pérdida de su presentación, entre otros casos.

En el Capítulo I expondremos la justificación de nuestro trabajo al igual que nuestros objetivos a alcanzar. Veremos el Estado del Arte donde conoceremos a los diferentes software más populares que ofrecen el mismo servicio que se pretende realizar mencionando las diferencias entre cada uno de ellos.

Posteriormente en el Capítulo II veremos la teoría y conceptos básicos para la realización del trabajo, profundizando en temas como HTML, tecnologías lado servidor, tecnologías lado cliente y el estándar ODF.

En los siguientes Capítulos III y IV se presentan la metodología a usar el análisis de nuestro trabajo y el diseño del mismo.

Finalmente en Capítulo V veremos el resultado final del trabajo, las pruebas que se realizaron a cada función de la aplicación, conocer la interfaz donde los usuarios interactúan y editan sus presentaciones.

## **1.2 Justificación**

En el mercado existe una gran variedad de software en línea para realizar presentaciones digitales mediante el uso de una interfaz muy similar entre ellos, sin embargo si queremos editar la presentación y no tenemos Internet, no podremos hacerlo y aunque algunos permiten descargar una aplicación de escritorio hay que pagar el software para su uso.

Para resolver el inconveniente de poder hacer ediciones de las presentaciones sin estar conectado y posteriormente tenerlas disponibles en línea o viceversa se plantea realizar este editor de presentaciones que permitirá editar, ejecutar o guardar una aplicación estando o no en línea.

El programa para desarrollar presentaciones será una aplicación Web para usarla en Internet pero también se abre la posibilidad a que el usuario pueda descargarlo de forma gratuita y usarlo como una aplicación de escritorio (fuera de línea).

Además de que la mayoría de las aplicaciones generan una película de flash para ejecutar. En nuestro trabajo se generará una página HTML con el código en javascript necesario para ejecutar la presentación sin necesidad de tener algún plug-in instalado.

El sistema le permitirá al usuario tener acceso a sus presentaciones donde quiera que este mediante una conexión a Internet, incluso si deja su computadora en casa, la pierde o le es robada, con esto sus presentaciones estarán siempre guardadas de manera segura en línea, todo lo que necesitará será un navegador y ahí estarán sus presentaciones. El sistema también le permitirá descargar sus presentaciones y abrirlas con otro software que lea el formato ODF haciendo que incluso sin una conexión a Internet el usuario pueda editar sus presentaciones.

## **1.3 Objetivos**

### **1.3.1 Objetivo General**

Desarrollar un software para realizar presentaciones digitales que permita realizar la edición de las presentaciones en línea y fuera de línea.

### **1.3.2 Objetivos Específicos**

- Crear la aplicación en línea y fuera de línea.
- Establecer el lenguaje XML que se utilizará en la edición de las diapositivas
- Utilizar de forma alterna el estándar ODF para hacer las presentaciones portables a Open Office Impress o Power Point.

## **1.4 Estado del Arte**

Nuestro sistema pretende ser una alternativa al software de línea y de escritorio comerciales dedicados a la realización de presentaciones digitales.

Actualmente hay algunos programas en línea populares al igual que aplicaciones de escritorio todos estos se comparan sus características en la tabla 2.a:

**Tabla 2.a. Tabla comparativa de software en línea de presentaciones digitales. [20]**

Aplicaciones	Tipo de aplicación	Descargable como aplicación de escritorio	Guarda datos en el servidor	Permite compartir las presentaciones	Auto recuperación de las presentaciones	Número máximo de diapositivas	Tamaño de almacenamiento	Soporte multimedia	Otros
Prezi [1]	Web	Si la versión de paga	Si	Si	No	No tiene un solo pizarrón	100 Mb	Si	Modo distinto de desarrollar presentaciones.
280 Slides [2]	Web	No	Si	Si	Si	280	-	Si	-
SlideRocket [3]	Web	Si la versión de paga	Si	Si	Si	Limitado a 200 Mb	200 Mb	Si	Trial de 14 días
Power Point	Escritorio	-	-	No	No	Sin limite	-	Si	Hay que contar con licencia para su uso

## CAPITULO II. Marco Teórico

### 2.1 Software en Línea

Son programas que funcionan desde el navegador, se utilizan sin descargarlos ni instalarlos en el equipo. Actualmente se utilizan mucho menos que las aplicaciones de escritorio, entre otros motivos porque ofrecen funcionalidades más limitadas que las aplicaciones de escritorio tradicionales; pero su uso está creciendo a medida que crecen las funcionalidades que se ofrecen desde los navegadores.

El software en línea no reside en nuestro equipo sino en un servidor de Internet al cual es posible conectarse desde un navegador, como a cualquier página Web. Este servidor es el que soporta toda (o gran parte) de la lógica de ejecución de la aplicación, el usuario utiliza el programa a través de su navegador, éste manda peticiones al servidor, quien las interpreta, ejecuta las tareas pertinentes y devuelve el resultado al navegador. [4]

Sus principales ventajas y desventajas son las que se enuncian en la tabla 2.a.

**Tabla 2.a Ventajas y desventajas del software en línea. [4]**

Ventajas	Inconvenientes
<ul style="list-style-type: none"><li>• El cliente no necesariamente debe tener un área especializada para soportar el sistema, por lo que baja sus costos y su riesgo de inversión.</li><li>• <b>Ahorra tiempo:</b> Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.</li><li>• <b>No hay problemas de compatibilidad:</b> Basta tener un navegador actualizado para poder utilizarlas.</li><li>• <b>No ocupan espacio</b> en nuestro disco duro.</li><li>• <b>Actualizaciones inmediatas:</b> Como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.</li></ul>	<ul style="list-style-type: none"><li>• Habitualmente ofrecen <b>menos funcionalidades</b> que las aplicaciones de escritorio. Se debe a que las funcionalidades que se pueden realizar desde un navegador son más limitadas que las que se pueden realizar desde el sistema operativo.</li><li>• <b>Si no se dispone de conexión a Internet</b> no se pueden usar.</li><li>• La <b>disponibilidad depende de un tercero</b>, el propietario de la aplicación. Así que la disponibilidad del servicio está supeditada al proveedor.</li><li>• La persona usuaria no tiene acceso directo a sus contenidos, ya que están guardados en un lugar remoto, y en caso de no contar con mecanismos de cifrado y control, disminuye el índice de privacidad, control y seguridad que ello supone, ya que la compañía TI podría consultarlos.</li></ul>

Continuación de la tabla 3.a. Ventajas y desventajas del software en línea

<ul style="list-style-type: none"><li>• <b>Consumo de recursos bajo:</b> Dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otro ordenador.</li><li>• <b>Multiplataforma:</b> Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.</li><li>• <b>Portables:</b> Es independiente de la computadora donde se utilice (una PC de escritorio, una laptop, un celular, etc.) porque se accede a través de una página Web (sólo es necesario disponer de acceso a Internet).</li></ul>	<ul style="list-style-type: none"><li>• El usuario no tiene acceso al programa, por lo cual no puede hacer modificaciones (dependiendo de la modalidad del contrato de servicios que tenga con la compañía TI).</li><li>• Al estar el servicio y el programa dependientes de la misma empresa no permite a la usuaria migrar a otro servicio utilizando el mismo programa (dependiendo de la modalidad del contrato de servicios con la compañía de TI).</li></ul>
--	--

## 2.2 Hiper Text Markup Language (HTML)

Definiéndolo de forma sencilla, *"HTML es lo que se utiliza para crear todas las páginas Web de Internet"*. Más concretamente, HTML es el *lenguaje* con el que se *"escriben"* la mayoría de páginas Web.

Los diseñadores utilizan el lenguaje HTML para crear sus páginas Web, los programas que utilizan los diseñadores generan páginas escritas en HTML y los navegadores que utilizamos los usuarios muestran las páginas Web después de leer su contenido HTML.

Aunque HTML es un lenguaje que utiliza los ordenadores y los programas de diseño, es muy fácil de aprender y escribir por parte de las personas.

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium, más conocido como W3C. Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo.

El propio W3C define el lenguaje HTML como *"un lenguaje reconocido universalmente y que permite publicar información de forma global"*. Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica. [5]

La última versión de HTML es la 5 que incluye novedades significativas en diversos ámbitos; no sólo se trata de incorporar nuevas etiquetas o eliminar otras, sino que supone mejoras en áreas que hasta ahora quedaban fuera del lenguaje y para las que se necesitaba utilizar otras tecnologías.

La adición de nuevos elementos es sin duda una noticia impactante, sobre todo después de ver como el desarrollo del lenguaje HTML lleva parado desde 1999 con el lanzamiento de HTML4. El consorcio mundial de la Web hasta ahora había estado centrado en el desarrollo de SGML, XML, XForms, RSS, CSS e incluso AJAX.

Existen nuevos elementos estructurales que van a dar mucho juego, por ejemplo a la hora de ver el código fuente de una página Web, ya no será lo mismo. Muchos elementos “div”, cuyo identificador era muy común repetir ahora tienen su propio elemento HTML5. Veamos por qué. [6]

- section: Puede ser un capítulo, una sección de un capítulo o básicamente cualquier cosa que incluya su propio encabezamiento.
- header: La cabecera de una página. No confundir con el elemento “head”
- footer: El final de la página.
- nav: Una colección de links a otras páginas
- article: Una entrada independiente en un blog, revista, etc.

Estos nuevos elementos nos harán dar un paso más hacia la Web semántica. Algunas páginas de hecho podrán cambiar del todo su código para incluir estas novedades: [6]

- Estructura del cuerpo: La mayoría de las aplicaciones Web tienen un formato común, formado por elementos como cabecera, pie, navegadores, etc. HTML 5 permite agrupar todas estas partes de una Web en nuevas etiquetas que representarán cada uno de las partes típicas de una página.
- Canvas: Es un nuevo componente que permitirá dibujar, por medio de las funciones de un API, en la página todo tipo de formas, que podrán estar animadas y responder a interacción del usuario. Es algo así como las posibilidades que nos ofrece Flash, pero dentro de la especificación del HTML y sin la necesidad de tener instalado ningún plug-in.
- Bases de datos locales: Se abre la posibilidad de tener una base de datos en el cliente y poder ejecutar consultas SQL (aún no está bien definido este apartado).
- Aplicaciones Web Off-line: Existirá otro API para el trabajo con aplicaciones Web, que se podrán desarrollar de modo que funcionen también en local y sin estar conectados a Internet.
- Nuevas APIs para interfaz de usuario: temas tan utilizados como el "drag & drop" (arrastrar y soltar) en las interfaces de usuario de los programas convencionales, serán incorporadas al HTML 5 por medio de un API.
- Historial de deshacer para todos los cambios realizados en el DOM.

### **2.3 Open Document Format (ODF)**

El Formato de Documento Abierto para Aplicaciones Ofimáticas de OASIS (en inglés, *OASIS Open Document Format for Office Applications*), también referido como OpenDocument u ODF, es un estándar de formato de fichero para el almacenamiento de documentos ofimáticos tales como hojas de cálculo, memorandos, gráficas y

presentaciones. Su desarrollo ha sido encomendado a la organización OASIS y está basado en un esquema inicialmente creado por OpenOffice.org. OASIS (acrónimo de Organization for the Advancement of Structured Information Standards) es un consorcio internacional sin fines de lucro que orienta el desarrollo, la convergencia y la adopción de estándares.

El estándar ODF fue desarrollado públicamente por un grupo de organizaciones, es de acceso libre, y puede ser implementado por cualquiera sin restricción. El formato Open Document pretende ofrecer una alternativa abierta a los formatos de documentos propiedad de Microsoft cuyos requisitos de licencia impiden su empleo a diversos competidores. La motivación principal para usar formatos estándar reside en que las organizaciones e individuos que lo hacen evitan la dependencia de un único proveedor de software, permitiéndoles cambiar de entorno informático si su proveedor actual desaparece del mercado o cambia su modelo de licencia en términos menos favorables para el cliente.

Open Document es el primer estándar para documentos ofimáticos implementado por distintos competidores, visado por organismos de estandarizaciones independientes y susceptibles de ser implementado por cualquier proveedor.

Open Document fue aprobado como un estándar OASIS el 1 de mayo de 2005; así mismo fue publicado en 2006 como estándar ISO 26300 (Internacional Organization for Standardization).

En poco más de un año ya ha sido adoptado oficialmente para documentos públicos por gobiernos nacionales o regionales de varios países del mundo: Francia, España, Dinamarca, Noruega, Bélgica, Reino Unido, Malasia, India, Indonesia, Sudáfrica, Brasil, Rusia, Corea, Argentina (provincia de Misiones), Estados Unidos (Estado de Massachusetts), y muchos otros están actualmente en proceso de transición: Holanda, Finlandia, Alemania, Japón. [7]

## **2.4 Tecnologías lado servidor**

Un lenguaje del lado servidor es independiente del navegador utilizado, que no necesitará plug-in especiales para visualizar correctamente cualquier página. Además, el código de los script puede ocultarse al cliente, que sólo verá el código HTML terminado. Por contra, la prueba y depuración de un script deberá hacerse desde el servidor.

### **2.4.1 *PHP Hypertext Pre-processor (PHP)***

PHP es un lenguaje de código abierto muy popular especialmente adecuado para desarrollo Web y que puede ser incrustado en HTML.

Lo que distingue a PHP es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá los resultados de ejecutar el script, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. El servidor Web puede ser incluso configurado para que procese todos los archivos HTML con PHP y entonces no hay manera que los usuarios puedan saber que tienes debajo de la manga.



Lo mejor de usar PHP es que es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales. Aunque el desarrollo de PHP está centrado en programación de scripts en lado servidor, se puede utilizar para muchas otras cosas.

Con PHP se puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. Y esto no es todo, se puede hacer mucho más.

PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. [8] PHP soporta la mayoría de servidores Web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd y muchos otros. PHP tiene módulos disponibles para la mayoría de los servidores, para aquellos otros que soporten el estándar CGI, PHP puede usarse como procesador CGI.

De modo que, con PHP tiene la libertad de elegir el sistema operativo y el servidor de su gusto. También tiene la posibilidad de usar programación orientada a objetos (POO). Aunque no todas las características estándar de la programación POO están implementadas en PHP 4, muchas bibliotecas y aplicaciones grandes están escritas usando íntegramente POO, PHP 5 soluciona los puntos débiles de POO en PHP 4 e introduce soporte completo para objetos.

Con PHP no se encuentra limitado a resultados en HTML. Entre las habilidades de PHP se incluyen: creación de imágenes, archivos PDF e incluso películas Flash sobre la marcha. También puede presentar otros resultados, como XHTML y cualquier otro tipo de ficheros XML. PHP puede auto-generar éstos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla, creando un caché en el lado servidor para contenido dinámico.

Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir un interfaz vía Web para una base de datos es una tarea increíblemente simple con PHP. Las siguientes bases de datos están soportadas actualmente:

- Adabas D
- dBase
- Empress
- IBM DB2
- ODBC
- InterBase
- FrontBase
- mSQL
- MySQL
- Oracle
- Entre otros (Informix, Ingres, Direct MS-SQL, HyperWave, Ovrimos, PostgreSQL, SQLite, Solid, Sybase, Velocis, Unix dbm )

También PHP tiene abstracción de base de datos (llamada PDO) que permite usar de forma transparente cualquier base de datos soportada por la extensión. Adicionalmente, PHP soporta ODBC (el Estándar Abierto de Conexión con Bases de Datos), así que puede conectarse a cualquier base de datos que soporte tal estándar.

PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También se pueden crear sockets puros e interactuar usando cualquier otro protocolo. PHP soporta WDDX para el intercambio de datos entre lenguajes de programación en Web. Y hablando de interconexión, PHP puede utilizar objetos Java de forma transparente como objetos PHP Y la extensión de CORBA puede ser utilizada para acceder a objetos remotos.

PHP tiene unas características muy útiles para el procesamiento de texto, desde expresiones regulares POSIX extendidas o tipo Perl hasta procesadores de documentos XML. Para procesar y acceder a documentos XML, PHP 4 soporta los estándares SAX y DOM y también Puede utilizar la extensión XSLT para transformar documentos XML. PHP 5 estandariza todas las extensiones XML en una sólida base de libxml2 y extiende las características establecer añadir SimpleXML y soporte para XMLReader. [9]

A continuación se presenta la tabla 3.b, donde se comparan algunos lenguajes que son interpretados en el servidor.

**Tabla 3.b Tabla comparativa de lenguajes del lado del servidor. [20]**

Lenguaje / Característica	JSP	PHP	ASP .NET
<b>Enfoque de programación</b>	Orientado a objetos, limpio, demasiado descriptivo.	Permite utilizar tanto POO como estructurada, funcional y rápido de programar, puede llegar a ser problemático.	Muy rápido de programar, enlaces ODBC a cualquier base de datos, el código puede ser trasladado fácilmente a otro lenguaje, nos amarra a Microsoft.
<b>Características de acceso a bases de datos.</b>	Descriptivo, usa la misma interfaz de programación para todas las bases de datos usando controladores JDBC.	Fácil de usar, muy buena integración con MySQL y PostgreSQL.	Estrechamente integrado a SQL server.
<b>Manipulación de datos</b>	Demasiado descriptivo, código orientado a objetos	Gran funcionalidad y fácil de codificar.	Gran funcionalidad y fácil de codificar.

Continuación de la tabla 3.b Tabla comparativa de lenguajes del lado del servidor.

<b>Características Web como: correo, sesiones, manejo de formularios e inclusiones.</b>	Complicado y requiere de mucho código.	Funcionalidad incorporada, funciones fáciles de usar escritas para tareas específicas.	Funciones fácil de usar escritas para tareas específicas.
<b>Extensibilidad</b>	Clases de Java y Bibliotecas. Se ejecuta en entorno limitado y de difícil aproximación a una interfaz nativa de java, necesaria para la integración con programas en el servidor.	PHP / C / Cualquier cosa ejecutable por el sistema operativo. Puede interactuar fácilmente con programas en el servidor. Muy buen soporte para código nativo.	Microsoft ha publicado algunos marcos de extensión que se conectan a ASP.NET y extender su funcionalidad

## 2.5 Tecnologías lado cliente

Un lenguaje del lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. Pero nuestra página no se verá bien si el ordenador cliente no tiene instalados los plug-in adecuados. El código, tanto del hipertexto como de los scripts, es accesible a cualquiera y ello puede afectar a la seguridad.

### 2.5.1 JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas Web dinámicas.

Una página Web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

Javascript es el siguiente paso, después del HTML, que puede dar un programador de Web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza.

Entre las acciones típicas que se pueden realizar en Javascript tenemos dos vertientes. Por un lado los efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro

dinamismo. Por el otro, javascript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo. [10]

En cualquier caso, es importante señalar que, pese a la similitud de nombres, JavaScript no es Java.

Las aplicaciones cliente y servidor en JavaScript comparten el mismo núcleo de lenguaje. Este núcleo se corresponde con ECMA-262, el lenguaje de scripts estándar de la Oficina de Estándares de la Unión Europea, con algunos añadidos extra. Aunque Javascript de cliente y de servidor comparten el mismo conjunto base de funciones y características; en algunos casos se utilizan de distinta forma. Los componentes de JavaScript son los siguientes:

- Núcleo de JavaScript (Core JavaScript).
- JavaScript para Cliente.
- JavaScript para Servidor.

JavaScript para cliente engloba el núcleo del lenguaje y algunos elementos adicionales como, por ejemplo, una serie de objetos predefinidos que sólo son relevantes para la ejecución de JavaScript en el contexto de un cliente Web.

Así mismo, JavaScript para servidor incluye también el núcleo de lenguaje y los objetos predefinidos y funciones necesarias para el correcto funcionamiento en el marco de un servidor.

El código JavaScript para cliente se integra directamente en páginas HTML y es interpretado, en su totalidad, por el cliente Web en tiempo de ejecución.

Puesto que con frecuencia es necesario ofrecer el mayor rendimiento posible, las aplicaciones JavaScript desarrolladas para servidores se pueden compilar antes de instalarlas en dichos servidores. [11]

### ***2.5.2 Cascading Style Sheets (CSS)***

CSS, es una tecnología que nos permite crear páginas Web de una manera más exacta. Gracias a las CSS somos mucho más dueños de los resultados finales de la página, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores.

Las Hojas de Estilo en Cascada se escriben dentro del código HTML de la página Web, solo en casos avanzados se pueden escribir en un archivo a parte y enlazar la página con ese archivo. [5]

El objetivo inicial de CSS, es separar el contenido de la forma, esto ya se cumplió con las primeras especificaciones del lenguaje. Sin embargo, el objetivo de ofrecer un control total a los diseñadores sobre los elementos de la página ha sido más difícil de cubrir. Las especificaciones anteriores del lenguaje tenían muchas utilidades para aplicar estilos a las aplicaciones Web, pero los desarrolladores aun continúan usando

trucos diversos para conseguir efectos tan comunes o tan deseados como los bordes redondeados o el sombreado de elementos en la página.

CSS 1 ya significó un avance considerable a la hora de diseñar páginas Web, aportando mucho mayor control de los elementos de la página. Pero como todavía quedaron muchas otras cosas que los diseñadores deseaban hacer, pero que CSS no permitía especificar, éstos debían hacer uso de otros métodos para el diseño. Lo peor de estos métodos es que muchas veces implica alterar el contenido de la página para incorporar nuevas etiquetas HTML que permitan aplicar estilos de una manera más elaborada. Dada la necesidad de cambiar el contenido, para alterar al diseño y hacer cosas que CSS no permitía, se estaba dando al traste con alguno de los objetivos para los que CSS fue creado, que era el separar por completo el contenido de la forma.

CSS 2 incorporó algunas novedades interesantes, que hoy ya utilizamos habitualmente, pero CSS 3 todavía avanza un poco más en la dirección, de aportar más control sobre los elementos de la página.

Así pues, la novedad más importante que aporta CSS 3, de cara a los desarrolladores de aplicaciones Web, consiste en la incorporación de nuevos mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas, sin tener que recurrir a trucos, que a menudo complicaban el código de las Web.[12]

Aquí hay una lista de las principales propiedades que son novedad en CSS 3: [13]

- Bordes (border-color, border-image, border-radius)
- Fondos (hacer capas con múltiples imágenes de fondo)
- Color (colores HSL, colores RGB, opacidad)
- Texto (rotura de palabras largas)
- Interfaz (resize)
- Selectores (por atributos)
- Otros (creación de múltiples columnas de texto, Web Fonts)

### **2.5.3 Frameworks**

Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

Los frameworks son diseñados con la intención de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. Sin embargo, hay quejas comunes acerca de que el uso de frameworks añade código innecesario y que la preponderancia de frameworks competitivos y complementarios significa que el tiempo que se pasaba programando y diseñando ahora se gasta en aprender a usar frameworks.

### 2.5.3.1 Características de los frameworks

Es necesario entender que un framework es una abstracción de código común que provee funcionalidades genéricas que pueden ser utilizadas para desarrollar aplicaciones de manera rápida, fácil, modular y sencilla, ahorrando tiempo y esfuerzo. Entonces, un framework es concreto y también “incompleto”. Concreto porque es, desde un punto de vista simple, un conjunto de componentes; incompleto, porque por sí mismos no pueden ser utilizados, ya que guían a la solución de problemas de programación recurrentes, pero por lo general, no son la solución específica completa.

En su mayoría, los frameworks javascript proveen componentes para:

- **Compatibilidad:** Agregan la posibilidad de escribir código javascript totalmente compatible con todos los navegadores y motores Javascript más utilizados. Esto aumenta la portabilidad y eliminan el gran problema de incompatibilidad entre navegadores y sus motores intérpretes javascript.
- **Comunicación asíncrona (AJAX).** Usando este acercamiento, es fácil utilizar XMLHttpRequest para manejar y manipular los datos en los elementos de un sitio, aumentando la interactividad y experiencia del usuario.
- **DOM.** Maximizan la capacidad de agregar, editar, cambiar, eliminar elementos de manera dinámica agregando librerías que facilitan usar DOM.
- **Validación de Formularios.** Permiten de una manera relativamente fácil validar campos dentro de uno o varios formularios. Esto, desde el punto de vista del desarrollador, simplifica y reduce el código para procesar dichos formularios, ya que los datos llegan previamente validados, reduciendo los errores de tipos de datos.
- **Efectos visuales.** Utilizando la manipulación de los elementos, se pueden crear efectos visuales y animaciones. Entre los efectos se encuentran: aparecer y desaparecer, redimensionamiento, mover y más.
- **Almacenamiento Client-side.** En adición provee funciones para leer y escribir cookies. También proveen una abstracción de almacenamiento que permite a las aplicaciones Web guardar datos del lado del cliente, persistente y de manera segura.
- **Manejo JSON.** Incrementa al máximo el manejo de datos, que pueden ser utilizados para presentar informaciones de manera dinámica y en tiempo de ejecución.
- **Manejo de Eventos.** Esta característica, permite reaccionar de una manera u otra dependiendo de las acciones del usuario.
- **Recibidores de Datos.** Permiten utilizar diferentes formatos de datos como XML, HTML, Texto, JSON, entre otros.
- **“Arrastrar y Soltar”.** Mejor conocido como Drag and Drop. Es una funcionalidad que brinda la posibilidad de arrastrar elementos dentro de una misma página que interactúe con el resto de los elementos.

Para realizar el trabajo se investigaron 3 frameworks, que son los más utilizados por los programadores Web, estos frameworks son Prototype Mootools y jQuery, que se detallarán en los siguientes temas.

### 2.5.4 *Ashynchronous JavaScript And XML (AJAX)*

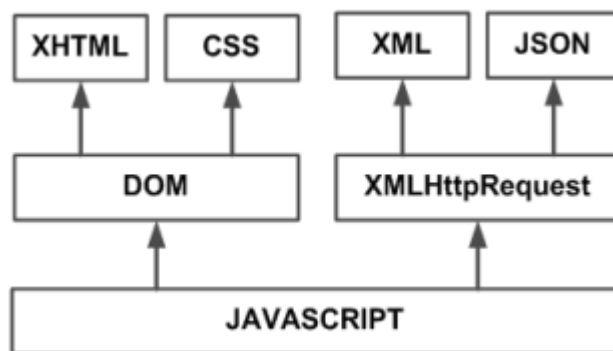
Antes de hablar de los frameworks, es necesario presentar el concepto AJAX, debido a que emplean AJAX.

AJAX no es una tecnología en sí mismo; en realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes, consiguiendo de esta manera una forma de navegación rápida, ágil y dinámica. Las tecnologías que forman a AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y manipulación de la información.
- XMLHttpRequest, para el intercambio asíncrono de la información.
- JavaScript, para unir todas las demás tecnologías.

En la figura 3.2 se muestra como están agrupadas estas tecnologías bajo el concepto AJAX. Desarrollar aplicaciones AJAX requiere un conocimiento de todas y cada una de las tecnologías mencionadas.

**Figura 3.2. Tecnologías agrupadas bajo el concepto AJAX.[14]**



En las aplicaciones Web tradicionales, las acciones del usuario en la página (oprimir un botón, seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador del usuario.

Esta técnica tradicional para crear aplicaciones Web funciona correctamente, pero no crea una buena sensación al usuario. Al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados. Si la aplicación debe realizar peticiones continuas, su uso se convierte en algo molesto.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor. [14]

### **2.5.5 Framework Prototype**

Este framework resulta muy interesante, pues hay muchos usuarios que lo utilizan habitualmente y con éxito. Parece una opción altamente profesional y además tiene la garantía que lo utiliza para la creación de sus aplicaciones Web empresas muy conocidas a nivel mundial. Pero hay ciertos detractores que acusan a este framework de ser muy pesado y ralentizar los sitios Web donde se utiliza.

Prototype es un framework que facilita el desarrollo de aplicaciones Web con JavaScript y AJAX. Su autor original es Sam Stephenson, aunque las últimas versiones incorporan código e ideas de muchos otros programadores. A pesar de que incluye decenas de utilidades, la librería es compacta y está programada de forma muy eficiente.

Prototype se ha convertido en poco tiempo en una referencia básica de AJAX y es la base de muchos otros frameworks. Las primeras versiones de Prototype no incluían ningún tipo de documentación, lo que dificultaba su uso y provocaba que la mayoría de usuarios desconocieran su verdadero potencial.

Afortunadamente, las versiones más recientes del framework disponen de una completa documentación de todas las funciones y métodos que componen su API. La documentación incluye la definición completa de cada método, sus atributos y varios ejemplos de uso. [14]

### **2.5.6 Framework Mootools**

Mootools es un framework javascript orientado a objetos que agrupa gran cantidad de herramientas y plug-in para facilitarles la vida a todos aquellos que a diario deban lidiar con efectos CSS, AJAX, JSON, etcétera sobre los diferentes navegadores.

Entendemos por framework a una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Los Frameworks son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. Quejas comunes acerca de que el uso de frameworks:

- Añade código innecesario
- El tiempo que se pasaba programando y diseñando ahora se gasta en aprender a usar frameworks.



Mootools esta en su versión 1.3 y al día de hoy, acumula muchísimas herramientas y plug-in realizados por una gran comunidad que se está formando. La librería completa, incluyendo plug-in y todo, pesa solo 42kb comprimida, pero dado que nos permiten descargar solo los componentes que vamos a utilizar, podemos hacerla adelgazar hasta 22kb sin perder sus herramientas más importantes.

Mootools es un framework Javascript, que sirve para crear fácilmente código Javascript independiente del navegador, de una forma rápida y directa. Mootools es un conjunto de librerías, también llamado API, que proveen clases de programación orientada a objetos en Javascript, para realizar una amplia gama de funcionalidades en páginas Web, como trabajo con capas, efectos diversos, AJAX y mucho más. [15]

Mootools es una opción interesante, bastante difundida y que tiene una serie de ventajas como:

- Ligerero: el framework no pesa demasiado en Kb y la carga de la página Web en el navegador.
- Modular: Mootools se compone de diversos módulos y podemos seleccionar los que vamos a utilizar para incorporarlos en nuestras páginas Web, dejando los otros para que no ocupen tiempo de descarga ni procesamiento.
- Libre de errores: podemos confiar en el sistema porque las herramientas de Mootools funcionan perfectamente, sin emitir errores en tiempo de ejecución.
- Soportado por una amplia comunidad: existen muchos desarrolladores que lo utilizan con éxito y han creado una serie de componentes adicionales ya listos para usar en las páginas Web, como calendarios, editores de texto, etc.
- Soporte POO: MooTools contiene un robusto sistema de creación y herencia de clases que se asemeja al de los lenguajes de programación orientados a objetos.

Mootools es gratuito y de código abierto, con licencia MIT, que permite usarlo y modificarlo en cualquier caso de uso, por último Mootools es compatible con los siguientes navegadores: [16]

- Safari 2 y 3
- Internet Explorer 6, 7 y 8
- Mozilla Firefox 1.5+, 2.0+, y 3.0+
- Opera 9.0+
- Camino 1.5+
- Google Chrome

### **2.5.7 Framework jQuery**

jQuery es un framework para el lenguaje Javascript, sirve para implementar una serie de clases (de programación orientada a objetos) que nos permiten programar sin preocuparnos del navegador con el que nos está visitando el usuario, ya que funcionan de exacta forma en todas las plataformas más habituales.

Así pues, este framework Javascript, nos ofrece una infraestructura con la que tendremos mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, con jQuery obtendremos ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax, etc. Cuando programemos Javascript con jQuery tendremos a nuestra disposición una interfaz para

programación que nos permitirá hacer cosas con el navegador que estemos seguros que funcionarán para todos nuestros visitantes. Simplemente debemos conocer las librerías del framework y programar utilizando las clases, sus propiedades y métodos para la consecución de nuestros objetivos.

Además, todas estas ventajas que sin duda son muy de agradecer, con jQuery las obtenemos de manera gratuita, ya que el framework tiene licencia para uso en cualquier tipo de plataforma, personal o comercial. Para ello simplemente tendremos que incluir en nuestras páginas un script Javascript que contiene el código de jQuery, que podemos descargar de la propia página Web del producto y comenzar a utilizar el framework.

El archivo del framework ocupa unos 56 Kb, lo que es bastante razonable y no retrasará mucho la carga de nuestra página (si nuestro servidor envía los datos comprimidos, lo que es bastante normal, el peso de jQuery será de unos 19 Kb). Además, nuestro servidor lo enviará al cliente la primera vez que visite una página del sitio. En siguientes páginas el cliente ya tendrá el archivo del framework, por lo que no necesitará transferirlo y lo tomará de la caché. Con lo que la carga de la página sólo se verá afectada por el peso de este framework una vez por usuario. Las ventajas a la hora de desarrollo de las aplicaciones, así como las puertas que nos abre jQuery compensan extraordinariamente el peso del paquete.

Existen varias frameworks que ofrecen soluciones similares que también funcionan muy bien, que básicamente nos sirven para hacer lo mismo. Como es normal, cada uno de los frameworks tiene sus ventajas e inconvenientes, pero jQuery es un producto con una aceptación por parte de los programadores muy buena y un grado de penetración en el mercado muy amplio, lo que hace suponer que es una de las mejores opciones. Además, es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework. Otra cosa muy interesante es la dilatada comunidad de creadores de plug-in o componentes, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar asuntos como interfaces de usuario, galerías, votaciones, efectos diversos, etc.

jQuery es una herramienta imprescindible para desarrollar todas estas cosas sin tener que complicarte con los niveles más bajos del desarrollo, ya que muchas funcionalidades ya están implementadas, o bien las librerías del framework te permitirán realizar la programación mucho más rápida y libre de errores. [17]

Algunas de las características más relucientes de jQuery son:

- Selección de elementos DOM
- Interactividad y modificaciones del árbol DOM
- Incluye un plug-in básico de XPath
- Soporte y manipulación para CSS 1-3
- Eventos, efectos y animaciones
- AJAX
- Utilidades varias como obtener información del navegador, operar con objetos y vectores

jQuery es compatible con los siguiente navegadores: [17]

- Mozilla Firefox 2.0+
- Internet Explorer 6+
- Safari 3+
- Opera 9+
- Google Chrome 1+

## **2.6 Base de datos**

### **2.6.1 Sistema Gestor de Base de Datos**

Un sistema gestor de base de datos (SGBD) consiste en la colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos normalmente denominada base de datos, contiene información relevante para el proyecto. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente. Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información. Además, los sistemas de bases de datos deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos de acceso sin autorización. Si los datos van a ser compartidos entre diversos usuarios, el sistema debe evitar posibles resultados anómalos. Dado que la información es tan importante en la mayoría de las organizaciones, los científicos informáticos han desarrollado un amplio conjunto de conceptos y técnicas para la gestión de los datos. [18]

### **2.6.2 MySQL**

La base de datos MySQL se ha convertido en la base de datos de código abierto más popular debido a su gran rendimiento constante, alta confiabilidad y facilidad de uso. Es muy utilizado por los desarrolladores Web y mencionando a las industrias que lo usan en sus sistemas como Yahoo, Alcatel -Lucent, Google, Nokia, YouTube y Zappos.com.

MySQL es de las base de datos más populares por su código abierto, y es una excelente opción para la generación de aplicaciones basadas en la pila LAMP (Linux, Apache, MySQL, PHP / Perl / Python).

MySQL se ejecuta en más de 20 plataformas, incluyendo:

- Linux
- Windows
- Mac OS X
- Solaris
- HP-UX
- IBM AIX

MySQL es muy utilizado en aplicaciones Web, como Drupal o phpBB, en todas las plataformas ya mencionadas. Su popularidad como aplicación Web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones Web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante adelantar monitoreos sobre el desempeño para detectar y corregir errores tanto de SQL como de programación. [19]

Las características principales de MySQL son:

- Es un gestor de base de datos. Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda.
- Es una base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.
- Es Open Source. El código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL para aplicaciones no comerciales.
- Es una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet.
- Existe una gran cantidad de software que la usa.

## **CAPITULO III. Análisis**

### **3.1 Metodología**

Se va a utilizar una metodología para el desarrollo de Web, denominada Ingeniería Web, que consta con los siguientes pasos:

Formulación, análisis, diseño (arquitectónico, navegación, interfaz, de base de datos), implementación y pruebas.

### **3.2 Formulación**

-¿Cuál es la motivación principal de la aplicación Web?

- El poder usar el Internet para proveer de un servicio útil como lo es realizar presentaciones digitales, desarrollando la aplicación Web usando las nuevas tecnologías como el html5 y proveerlo de forma totalmente gratuita.

-¿Por qué es necesaria la aplicación Web?

- Para poder tener acceso a un editor de presentaciones donde quiera que se encuentre, contando con un navegador de Internet.

-¿Quién va a utilizar la aplicación Web?

- Cualquier usuario con la necesidad de crear o cargar una presentación, editándola en línea desde cualquier navegador de Internet.

-El sistema contará con una base de datos para que el usuario guarde sus presentaciones o si lo desea podrá compartir sus presentaciones con los demás usuarios del sistema.

### **3.3 Análisis**

#### **3.3.1 Análisis de contenido**

El sistema ofrecerá a los usuarios que puedan crear, modificar, visualizar, exportar y compartir sus presentaciones cuando guste y ya sea en línea o fuera de línea.

#### **3.3.2 Análisis de interacción**

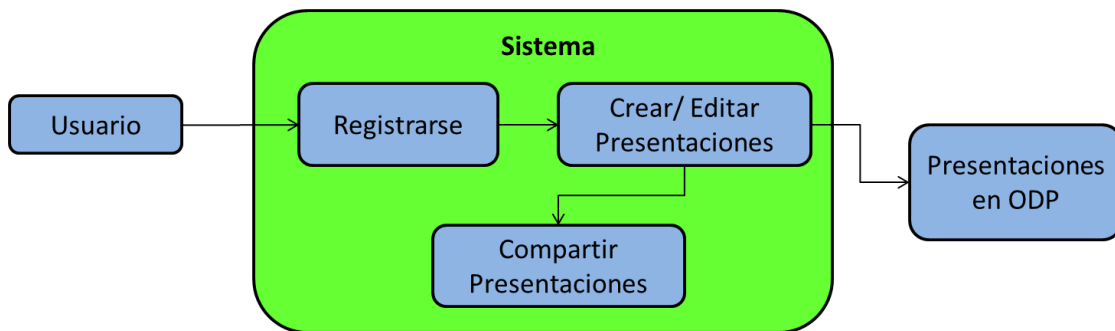
El usuario contará con diferentes funciones para el desarrollo de sus presentaciones, como lo son el diferente uso de plantillas y estilos, el uso de multimedia (imágenes, video, sonido), atributos para el texto, exportación de sus presentaciones así como el poder compartirlas con otros usuarios.

### 3.3.3 Análisis funcional

En este apartado mostraremos algunos diagramas en donde se definen las operaciones en el contenido y operaciones de procesamiento.

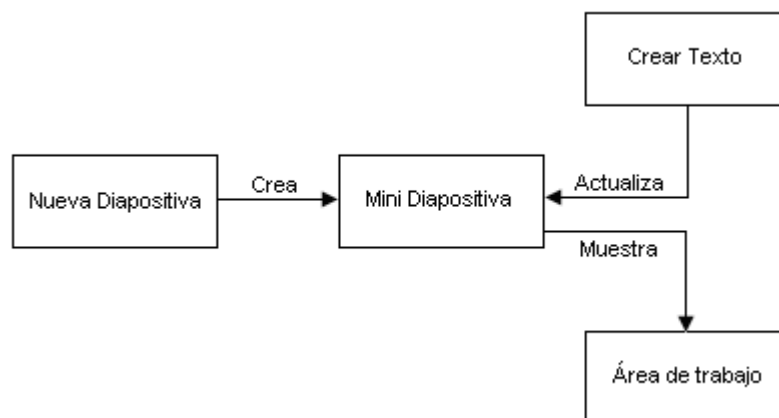
En el diagrama 3.a vemos la arquitectura básica del diseño.

**Diagrama 3.a Arquitectura básica del sistema. [20]**



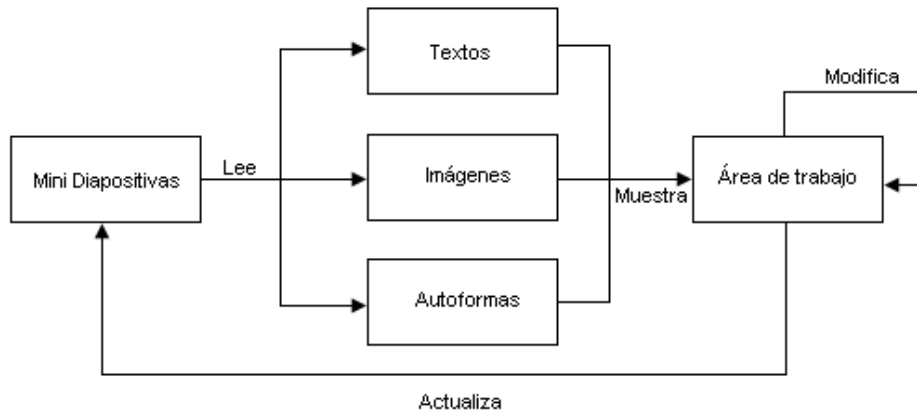
En el diagrama 3.b, observamos como será el procedimiento de crear un nuevo texto. Este proceso se inicia creando una nueva diapositiva, la cual crea una mini diapositiva, en la cual se hace el trabajo ya que al tenerla y ejecutar crear un texto se actualiza la mini diapositiva con una nueva área de texto guardando los datos de esa área de texto como son dimensiones, estilo y contenido, como el tipo de letra, tamaño de letra, color de fondo, etc.; finalmente la mini diapositiva muestra todo esto en el área de trabajo. Por lo que en la mini diapositiva quedan guardados todos los datos del área de trabajo, este proceso es igual para las autoformas e imágenes.

**Diagrama 3.b Creación de un texto. [20]**



En nuestro siguiente diagrama 3.c explicaremos una función principal que es la de ver en nuestra área de trabajo la diapositiva seleccionada.

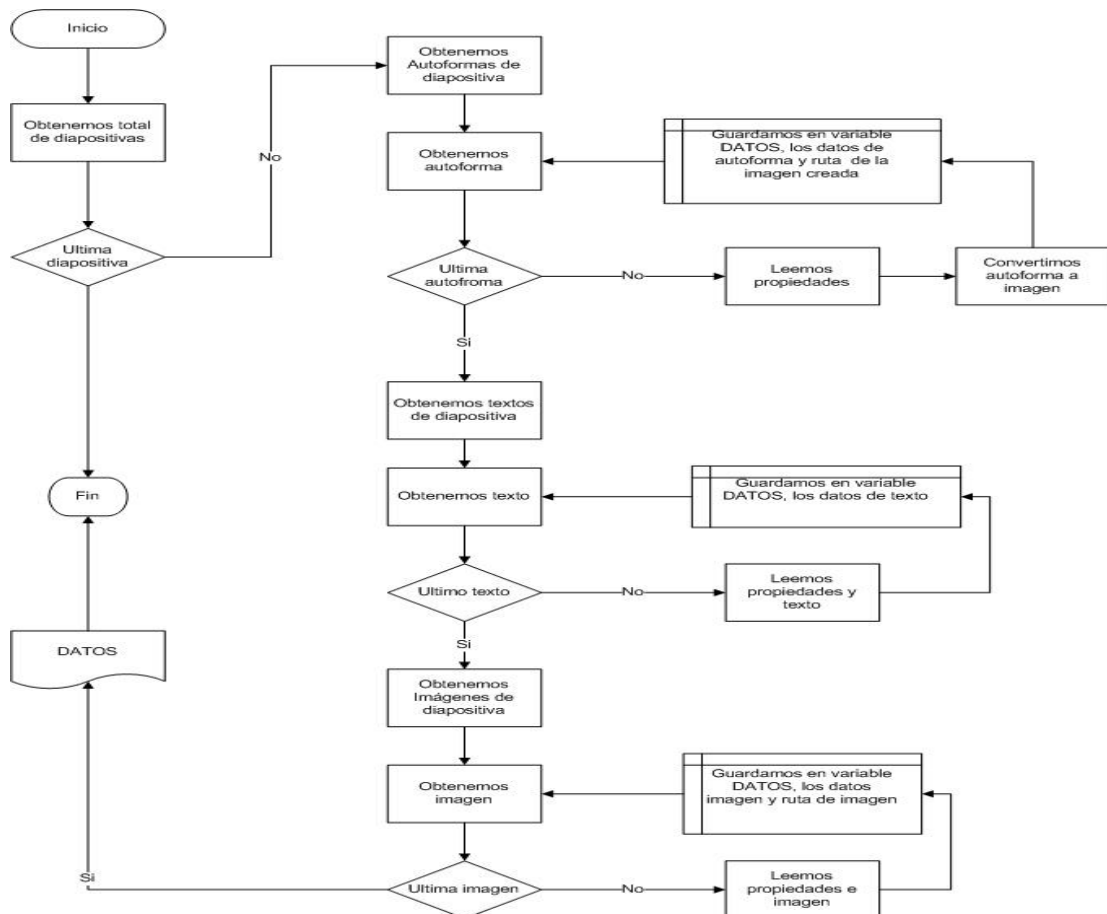
**Diagrama 3.c Mostrar diapositiva seleccionada. [20]**



Para poder visualizar en el área de trabajo las mini diapositivas son esenciales ya que en ellas se guarda todo el contenido ya sea los textos, imágenes y autoformas. Al seleccionar una mini diapositiva se estará leyendo los datos que contiene esa diapositiva; cuantos textos, imágenes, autoformas tiene sus estilos, color, tamaño, posición. Todos estos datos ya leídos se muestran automáticamente en el área de trabajo para su edición si es que desea modificar la diapositiva.

El diagrama de flujo 3.d muestra el proceso para guardar una presentación.

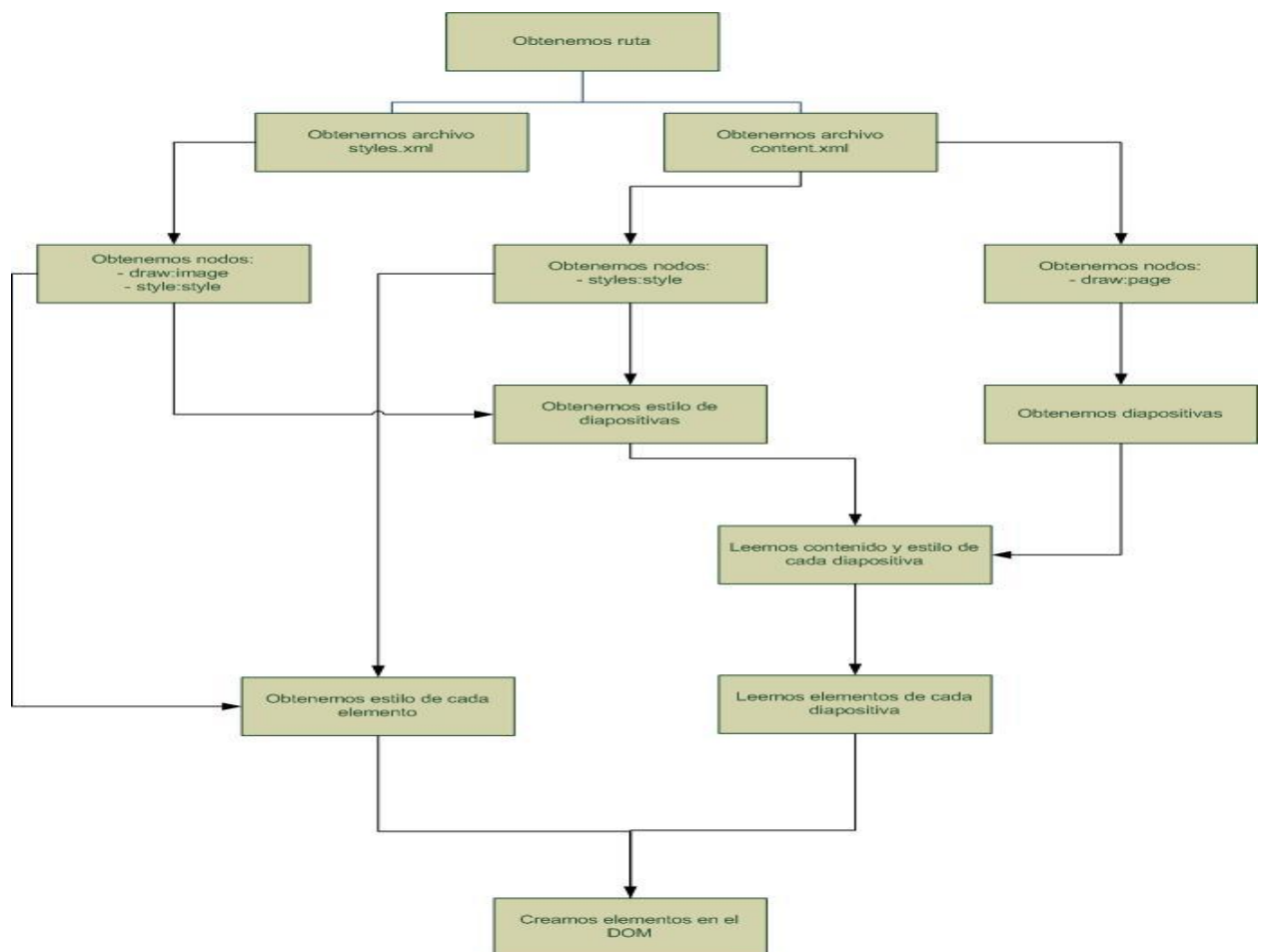
**Diagrama 3.d Guardar presentación. [20]**



Para realizar el proceso de guardar una presentación mostrado en el diagrama 3.d, obtenemos el total de las diapositivas y verificamos que no sea la ultima si es así, obtenemos las autoformas de la diapositiva si existen obtenemos autoforma, leemos sus propiedades, la convertimos en imagen y la guardamos en un variable, también se guarda la ruta de la imagen creada, al guardar las autoformas procedemos a guardar los textos, es un proceso igual obtenemos todos los textos y se guarda uno por uno leyendo sus propiedades y texto para posteriormente guardarlos. Por ultimo hacemos lo mismo para las imágenes y al final de tener todo el contenido de la diapositiva guardamos todo en la variable “DATOS”.

Para tener la lectura de un archivo ODP, este es un archivo comprimido, después de obtener la ruta del archivo obtenemos del “zip” los archivos “styles.xml” y “content.xml” con estos dos archivos en conjunto obtenemos los estilos de las diapositivas y el estilo de cada elemento. El archivo “content.xml” también contiene las diapositivas, ya teniendo esto junto con su estilo se lee el contenido y estilo de cada diapositiva y de ahí pasamos a cada elemento de la diapositiva con su respectivo estilo y finalmente creamos los elementos en el DOM. Todo esto se visualiza en el diagrama a bloques 3.e.

**Diagrama 3.e Lectura de un archivo ODP. [20]**





### 3.3.4 Análisis de configuración

#### 3.3.4.1 Requerimientos Funcionales

Los requerimientos funcionales de un sistema describen la funcionalidad o los servicios que se espera que éste provea y se exponen en la tabla 3.1

**Tabla 3.1 Requerimientos funcionales. [20]**

	Descripción del requerimiento
RF-1	El sistema permitirá la elaboración y edición de diapositivas las cuales se guardarán en formato ODP. El sistema también permitirá la lectura de archivos ODP no creados en el sistema.
RF-2	El sistema deberá permitir la creación de diapositivas que cumplan con los siguientes sub-requerimientos: RF-2.1 Inserción y edición de texto. RF-2.2 Inserción y eliminación de elementos multimedia.
RF-3	El sistema deberá tener una versión para descarga la cual deberá poder trabajar sin necesidad de una conexión a Internet.
RF-4	El usuario debe proporcionar datos tales como: su correo electrónico, un usuario y contraseña para registrarse y posteriormente poder tener acceso al sistema.
RF-5	El usuario debe realizar la diapositiva, el sistema no genera diapositivas automáticamente.
RF-6	El sistema deberá proporcionar una nueva contraseña en caso de que este la solicite. Esta será enviada al correo electrónico proporcionado al momento de registrarse.
RF-7	El sistema deberá presentar las presentaciones guardadas anteriormente por el usuario y ser capaz de editarlas.
RF-8	El sistema deberá guardar las presentaciones del usuario en línea si este así lo desea.
RF-9	El usuario deberá poder descargar la presentación que elaboro en línea.

#### 3.3.4.2 Requerimientos No funcionales

Son restricciones de los servicios o funciones ofrecidas por el sistema, ilustradas en la tabla 3.2, que a continuación se muestra:

**Tabla 3.2 Requerimientos no funcionales. [20]**

	Descripción del requerimiento
RNF-1	El sistema tanto en su versión con conexión como en la versión sin conexión solo podrá ser bien visualizado en navegadores que soporten HTML5.
RNF-2	El sistema deberá ofrecer una interfaz que implemente un conjunto de elementos gráficos y descriptivos para poder llevar a cabo las funciones requeridas en los requerimientos funcionales.
RNF-3	La versión off-line
RNF-4	El sistema podría no leer correctamente los archivos ODP generados por editores diferentes a Wediapz u Open Office.

Continuación de la tabla 3.2 Requerimientos no funcionales

<b>RNF-5</b>	El sistema no contiene un corrector ortográfico.
<b>RNF-6</b>	El usuario deberá proporcionar texto, imágenes y video a la hora de realizar o editar una diapositiva ya que el sistema no cuenta con una biblioteca multimedia.

### 3.3.4.3 Análisis de Factibilidad Operativa

Para describir este apartado nos haremos la pregunta: ¿Cubrirá el sistema las necesidades de un usuario final?, y la responderemos a continuación.

*La respuesta en una sola palabra sería si, porque creará diapositivas de manera dinámica y fácil de entender utilizando una interfaz parecida a otros sistemas de este tipo. Además de que guardara las presentaciones en un formato estandarizado el cual podrá ser interpretado por software como Open office. Y para un usuario con conocimientos de programación le será fácil modificar el código del sistema si este lo desea para mejorar uno o varios aspectos de este.*

### 3.3.4.4 Análisis de Factibilidad Técnica

Análisis de lenguajes y herramientas de software a emplear.

Descripción de los principales lenguajes utilizados para la programación de una página Web, que se van a utilizar las encontramos en la tabla 3.3

**Tabla 3.3 Descripción de lenguajes funcionales. [20]**

Lenguaje	Descripción
<b>HTML</b>	HTML es el lenguaje de marcado predominante para la elaboración de páginas Web. Es usado para describir la estructura y apariencia ella.
<b>JavaScript</b>	Es un lenguaje de scripting orientado a objetos, basado en prototipos, sin tipo y liviano, utilizado para acceder a objetos en aplicaciones. Se utiliza integrado en un navegador Web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas Web dinámicas.
<b>CSS ( hojas de estilo en cascada)</b>	Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). La idea que se encuentra detrás del desarrollo de CSS es separar la <i>estructura</i> de un documento de su <i>presentación</i> .

La siguiente tabla 3.4 muestra una comparativa de lenguajes para poder realizar un programa que nos permita realizar un guardado en el formato ODP.

**Tabla 3.4 Tabla comparativa de lenguajes de alto nivel. [20]**

Lenguaje / Característica	C#	Java	GTK	VB
Orientado a objetos	Si	Si	No	Si
Multiplataforma	No	Si	Si	No
Manejo de excepciones	Si	Si	No	Si
Librerías para el manejo de archivos ODP	No	Si	No	No
Precio Licencia	\$2799 dls.	Gratuito	Gratuito	\$2799 dls.

Se presenta en la tabla 3.5 una comparativa de algunos de los entornos de desarrollo (IDE's) más populares y versátiles para el desarrollo de aplicaciones Web.

**Tabla 3.5 Tabla comparativa de IDE's. [20]**

IDE / Característica	Netbeans	Eclipse	Dreamweaver CS5	RadPHP XE
Precio	Gratuito	Gratuito	\$399 dls.	\$299 dls.
Desarrollo en HTML	Si	Si (descargando complemento)	Si	Si
Desarrollo en JavaScript	Si	Si (descargando complemento)	Si	Si
Desarrollo en PHP	Si	Si (descargando complemento)	Si	Si

### 3.3.4.5 Análisis de Factibilidad Económica

Este apartado está orientado en determinar los costos monetarios de este proyecto de desarrollo de software, hemos incluido para tal fin una lista con los principales gastos a realizar:

Licencias de software:

**Tabla 3.6 Tabla de costos de software. [20]**

Aplicación	Versión	Costo
HTML	5	Gratuito
JavaScript	1.8.5	Gratuito
PHP	5.3.1	Gratuito
MySQL	5.1.52	Gratuito
Java	Versión 6 Update 16	Gratuito
Netbeans	6.9	Gratuito

Servidor:

**Tabla 3.7 Tabla de costos por uso de servidor. [20]**

Servidor	Costo	Tiempo
Dominio	\$1000	1 año

### **3.3.4.6 Conclusión de factibilidad**

HTML será usado para describir la estructura y el contenido de la interfaz del sistema. También será base para la inclusión de JavaScript el cual será utilizado para el manejo de una interfaz dinámica y fácil de entender para el usuario.

PHP será utilizado para la conexión con la base de datos y tendrá el control de la interacción del usuario con el servidor. Java será utilizado para generar un programa, el cual será ejecutado desde PHP, que nos permita guardar en el formato ODP.

Se tomó la decisión de emplear dichos lenguajes debido a que su uso es gratuito, lo cual no implica costo adicional en el sistema; además de lo anterior, los integrantes del proyecto tenemos mayor experiencia en la programación con PHP y Java que con otros lenguajes, por lo que no será necesario destinar tiempo de aprendizaje en este aspecto.

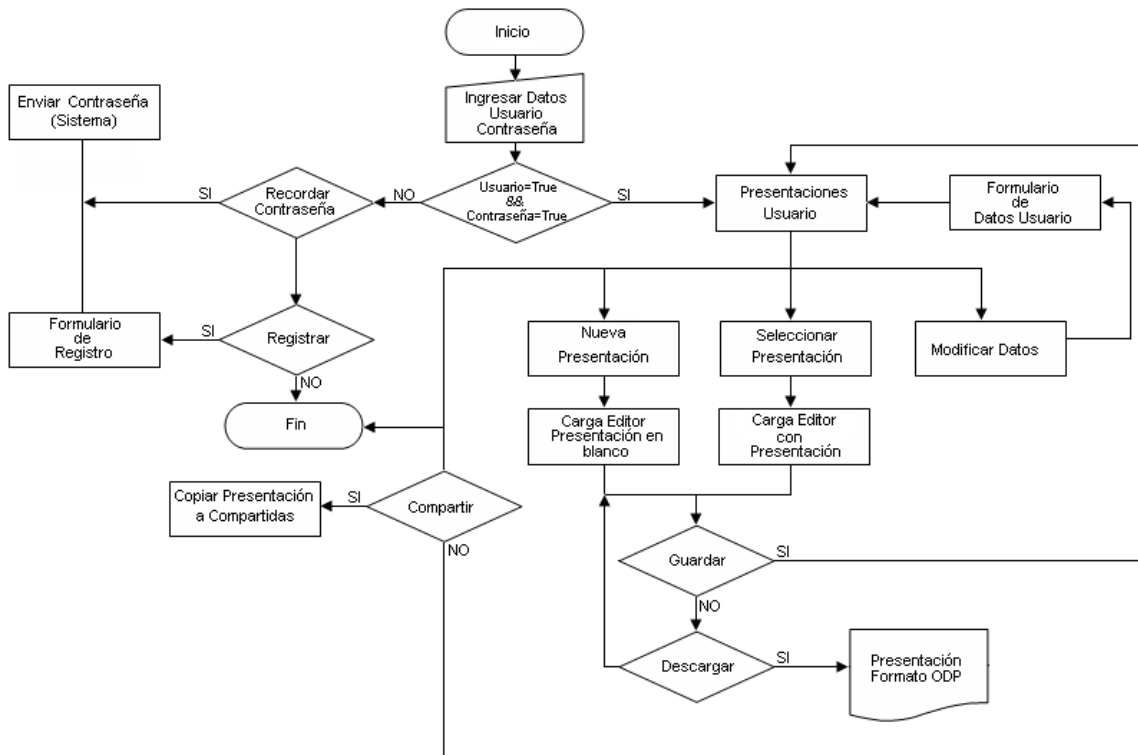
Y utilizaremos Netbeans para la escritura de código en todos estos lenguajes, debido a que es gratuito y cuenta con gran documentación.

## CAPITULO IV. Diseño

### 4.1 Diseño Arquitectónico

En el diseño arquitectónico definiremos las relaciones entre los elementos estructurales (módulos) del programa, para ello utilizaremos el diagrama de flujo. El diagramas de flujo esta representado en el diagrama 4.a.

**Diagrama 4.a Diagrama de flujo del sistema. [20]**



El diagrama 4.a nos muestra como esta relacionado nuestro sistema, al iniciar ingresamos los datos de usuario, el sistema verifica si son validos los campos de usuario y contraseña, si no son validos el sistema le dirá al usuario si desea recordar su contraseña si es así el sistema pedirá su usuario (correo electrónico) para mandarle un correo con su contraseña, si el usuario es nuevo y desea registrarse se le manda al formulario de registro.

Si el usuario y contraseña son validos, él podrá acceder a sus presentaciones con las opciones de crear una nueva presentación, seleccionar una presentación existente o modificar sus datos. Si el usuario desea una nueva presentación se cargará el editor para realizarla, si desea editar una presentación tendrá que seleccionar su presentación y se cargará en el editor para su modificación.

Al finalizar de editar una presentación ya sea nueva o una cargada, al momento de guardar si lo desea el usuario su presentación será guardada en su carpeta de presentaciones y ahí también puede elegir compartir su presentación al hacer esto los demás usuarios solo podrán ver su presentación mas no editarla.

Si el usuario descarga su presentación está será guardada en su equipo con formato ODP.

## 4.2 Estructura de una diapositiva

Una diapositiva va a contener información indispensable para que sea más fácil leerla y manejar sus datos. La diapositiva contará con sus propiedades que son:

- Número de diapositivas
- Número de textos
- Número de imágenes
- Número de autoformas
- Plantilla
- Fondo

Al insertar una autoforma, está crea sus propiedades que son:

- Identificador
- Tamaño
- Color de fondo
- Color de borde
- Posición

El tamaño y la posición de la autoforma serán a escala y la autoforma será un elemento “canvas”. Otro elemento que puede tener una diapositiva es una imagen que contará con las siguientes propiedades:

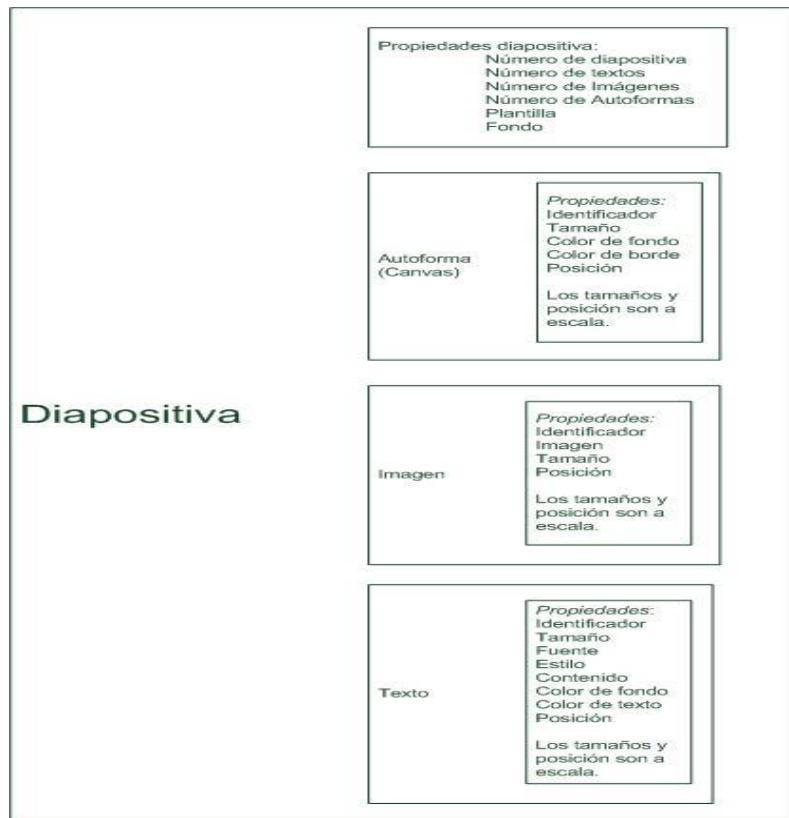
- Identificador
- Imagen
- Tamaño
- Posición

Al igual que la autoforma el tamaño y posición son a escala. Por ultimo la diapositiva también contiene texto y sus propiedades serán:

- Identificador
- Tamaño
- Fuente
- Estilo
- Contenido
- Color de fondo
- Color de texto
- Posición

De igual manera el tamaño y posición son a escala, está estructura es creada para manejar los datos de la presentación; todo lo anterior mencionado se ve en la figura 4.1

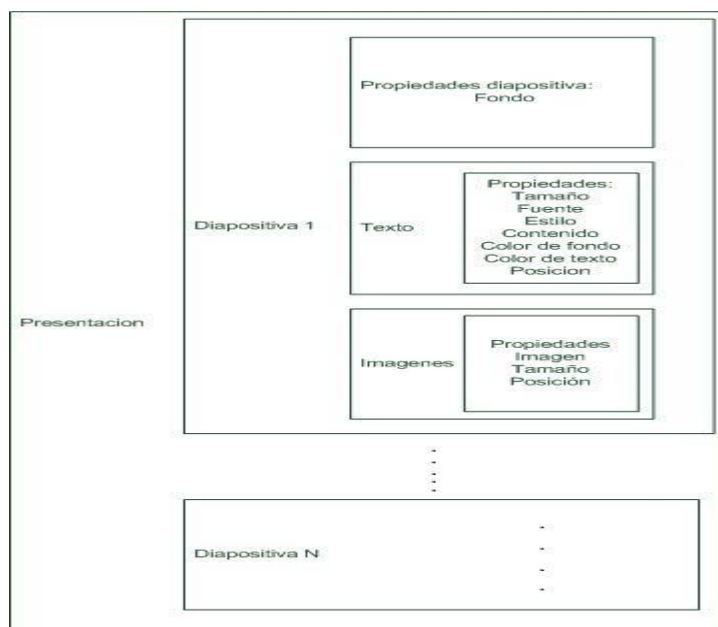
**Figura 4.1 Estructura de una diapositiva. [20]**



### 4.3 Estructura de un archivo

En este apartado veremos como estará estructurado un archivo o presentación, esto lo vemos mejor en la figura 4.2

**Figura 4.2 Estructura de un archivo. [20]**

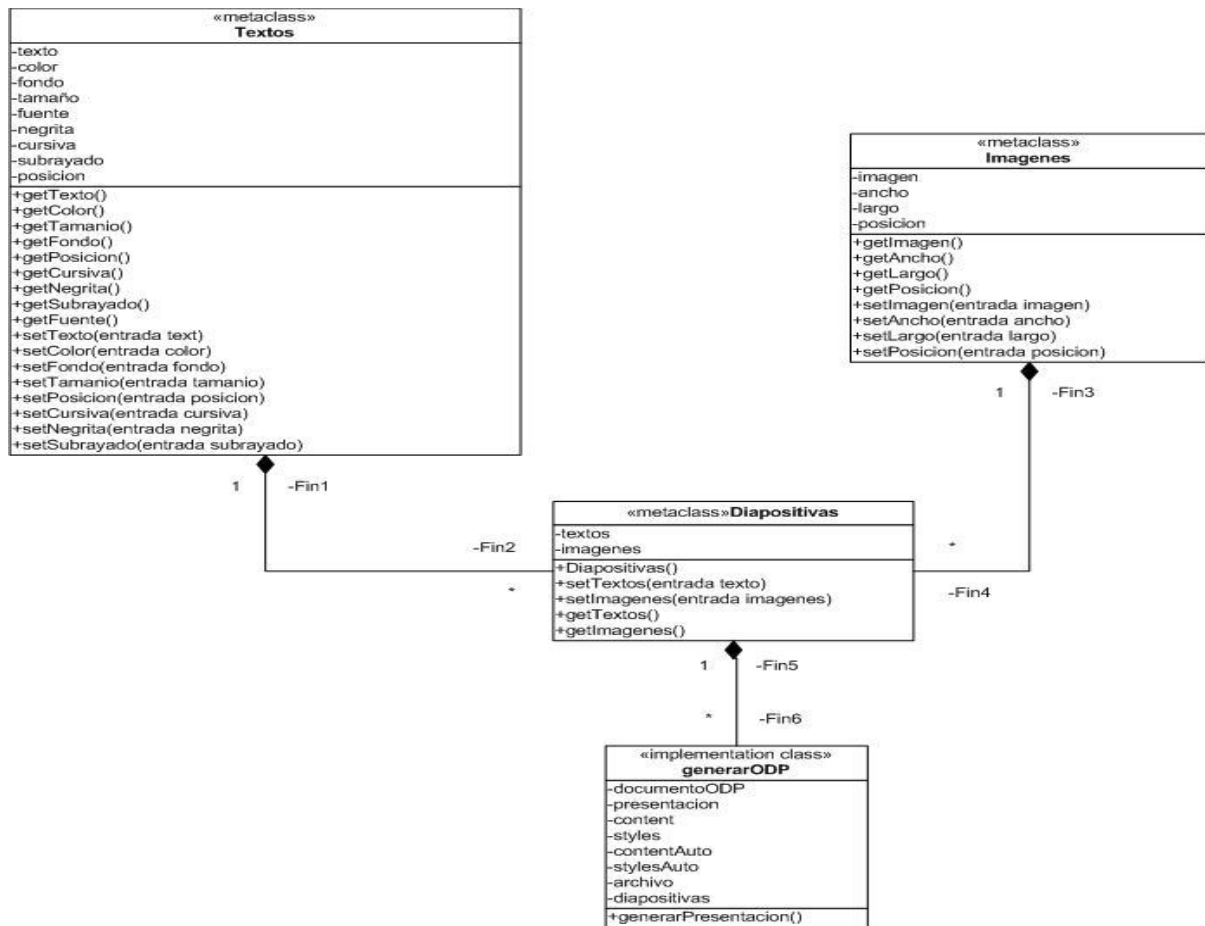


En la figura 4.2 vemos como una presentación esta conformada por varias diapositivas que a su vez contiene las propiedades de una diapositiva vistas anteriormente.

#### 4.4 Generar ODP

Para poder generar el ODP empleamos Java ya que tiene librerías para el manejo del formato ODP, el código a implementar será llamado con PHP, esta función a realizar se ve en el diagrama de clases 4.b

**Diagrama 4.b Diagrama de clases para generar ODP**



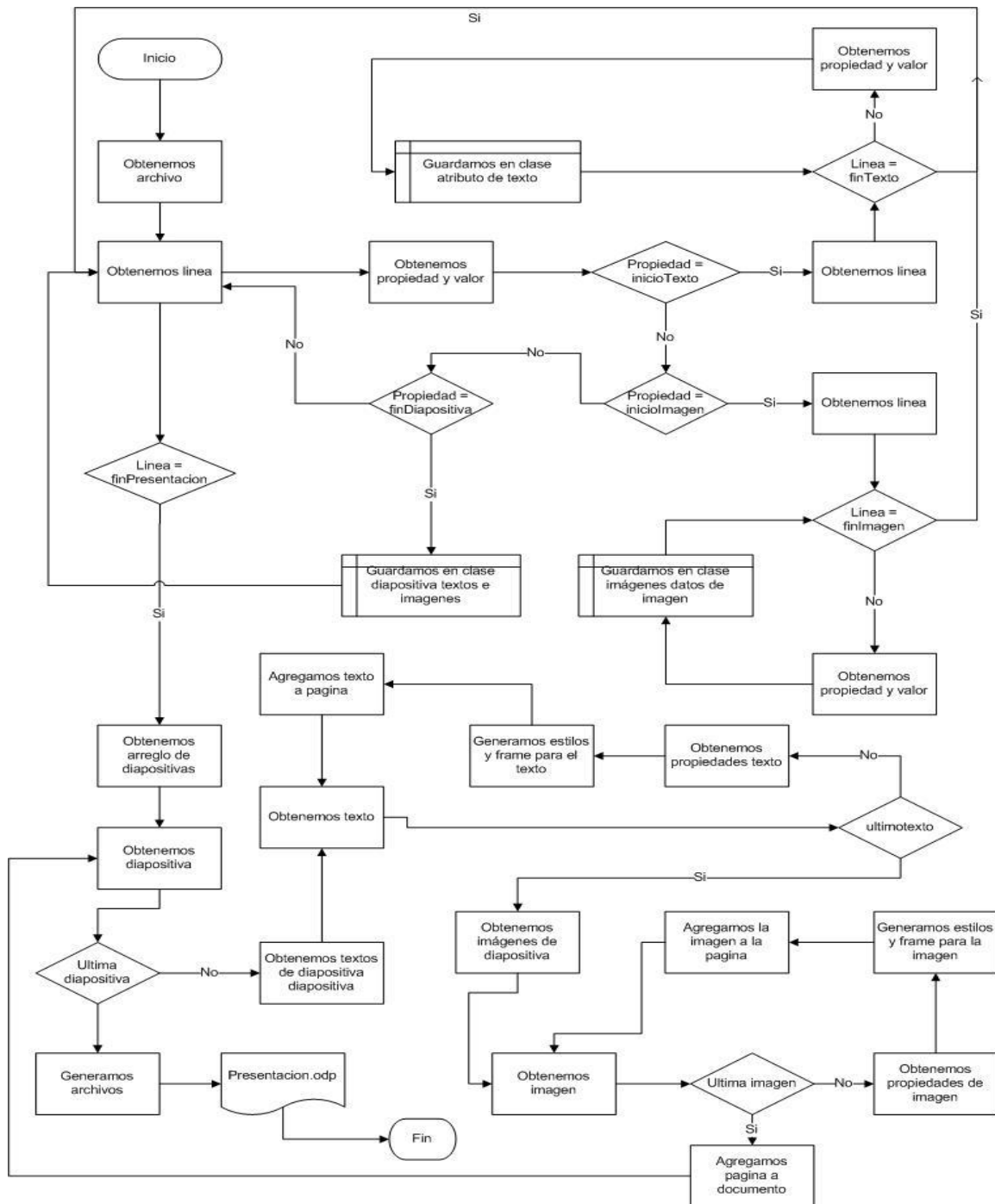
En el diagrama 4.b vemos que existe una clase “generadorODP” que tiene diferentes atributos que se ven en el diagrama 4.b, todos sus atributos son privados, y cuenta con el método “generarPresentacion” para poder crear una presentación en el formato ODP. Esta clase tiene una relación con la clase “Diapositivas”, su relación es por valor y cuenta con atributos privados que son texto e imágenes y posee sus respectivos métodos “set” y “get” de sus atributos. La clase “Diapositivas” tiene relación con otras dos clases, la clase “Imágenes” y la clase “Textos”, con sus atributos y métodos “set” y “get” de cada atributo visualizados en el diagrama 4.b.

En el siguiente diagrama 4.c vemos todo el proceso para generar el ODP, como primer paso obtenemos el archivo (presentación), obtenemos línea y creamos un ciclo para pasar por toda la presentación y estar guardando en las clases “Imágenes” y “Textos” todo el contenido de textos e imágenes en cada diapositiva al concluir el ciclo de haber guardado



todos los textos e imágenes, guardamos todo en la clase “Diapositivas”. El siguiente paso es obtener el arreglo de diapositivas y un ciclo mas diapositiva por diapositiva obtenemos los textos, sus propiedades, generamos estilos y frame para el texto y agregamos el texto a una pagina, al acabar con los textos en una diapositiva pasamos a las imágenes obteniendo igual sus propiedades, se genera el estilo y frame y se agrega a la pagina. Al concluir con la ultima imagen agregamos la pagina al documento y pasamos a la siguiente diapositiva; al llegar a la ultima diapositiva generamos los archivos y obtenemos como salida una presentación en formato ODP.

**Diagrama 4.c Diagrama de Flujo para generar ODP**



## 4.5 Diseño de Interfaz

La interfaz de la aplicación Web es la primera impresión para el usuario. En el cual consideraremos los siguientes criterios de diseño:

- Probabilidad de errores con el servidor
- Codificación en lenguaje HTML, JavaScript
- Evitar el desplazamiento en la pantalla
- Equilibrio entre estética y funcionalidad

Como parte del diseño del sistema se diseñaron las correspondientes interfaces del sistema para que el usuario pueda interactuar con él, las interfaces están divididas en tres partes:

LogIn a Wediapz

- Entrar
- Registrarse
- Restablecer contraseña

Presentaciones de usuario

- Nueva presentación
- Buscar presentaciones
- Editar presentación

Editor Wediapz

- Guardar presentación
- Cargar presentación
- Subir fotos y videos

### 4.5.1 Interfaz LogIn Wediapz

La interfaz LogIn está compuesta por un formulario principal, por un formulario para el registro de usuarios y un formulario para restablecer la contraseña de los usuarios.

**Figura 4.3 Formulario principal de LogIn. [20]**

Entrar Registrarse

Bienvenido

Introduce tu usuario y contraseña.

E-mail

Contraseña

¿Olvidaste tu contraseña?

Entrar

¿No eres usuario aún?  
[Regístrate ahora.](#)

© 2010 ESCOM TTR.

En la figura 4.4 se muestra el formulario de registro de usuarios donde se piden los datos como nombre, e-mail y contraseña.

**Figura 4.4 Formulario de registro de usuarios. [20]**

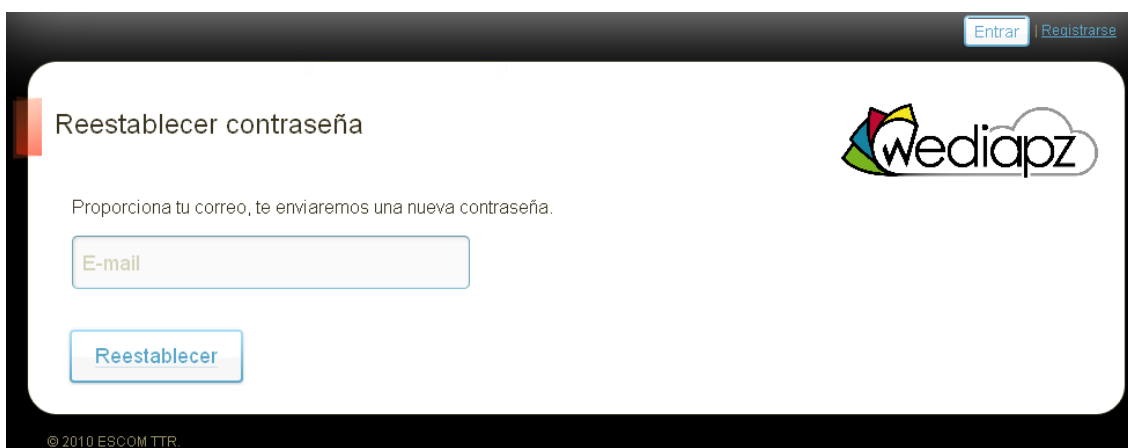


The screenshot shows a registration form titled "Forma de registro" for the WEDIAPZ website. At the top right, there are links for "Entrar" and "Registrarse". The form includes the following fields and elements:

- Logo for WEDIAPZ.
- Text: "Campos obligatorios \*".
- Input field for "Nombre(s)\*".
- Input field for "Apellidos".
- Input field for "E-mail\*".
- Text: "Nota: Tu e-mail será tu nombre de usuario."
- Input field for "Contraseña\*".
- Input field for "Confirmar\*".
- A button labeled "Registrarse y Continuar".
- Copyright notice: "© 2010 ESCOM TTR."

En la figura 4.5 vemos la ventana que aparecería si a un usuario olvida su contraseña, tan solo pidiendo el e-mail del usuario, el sistema le enviara un correo con su contraseña.

**Figura 4.5 Formulario para reestablecer contraseña del usuario. [20]**



The screenshot shows a password reset form titled "Reestablecer contraseña" for the WEDIAPZ website. At the top right, there are links for "Entrar" and "Registrarse". The form includes the following elements:

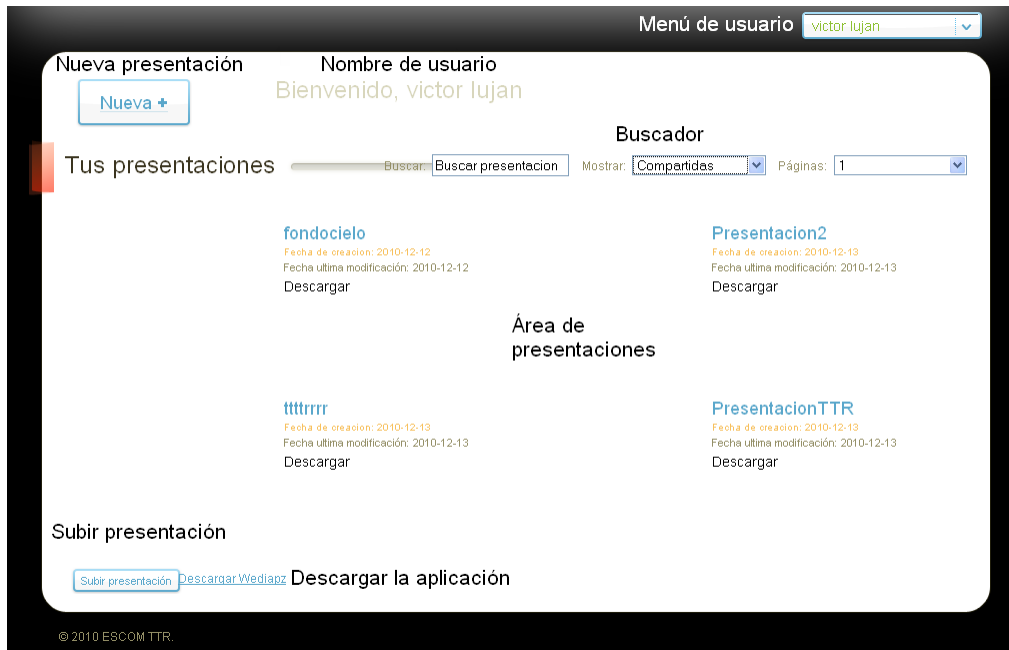
- Logo for WEDIAPZ.
- Text: "Proporciona tu correo, te enviaremos una nueva contraseña."
- Input field for "E-mail".
- A button labeled "Reestablecer".
- Copyright notice: "© 2010 ESCOM TTR."

#### 4.5.2 Presentaciones de usuario

En esta interfaz que vemos en la figura 4.6 se muestran las presentaciones hechas por el usuario y las presentaciones compartidas por otros usuarios y con la opción de crear una nueva diapositiva o subir una diapositiva con formato ODP, también hay un menú de

usuario para cerrar su sesión o configurar su cuenta. En esta misma interfaz esta para descargar la aplicación fuera de línea.

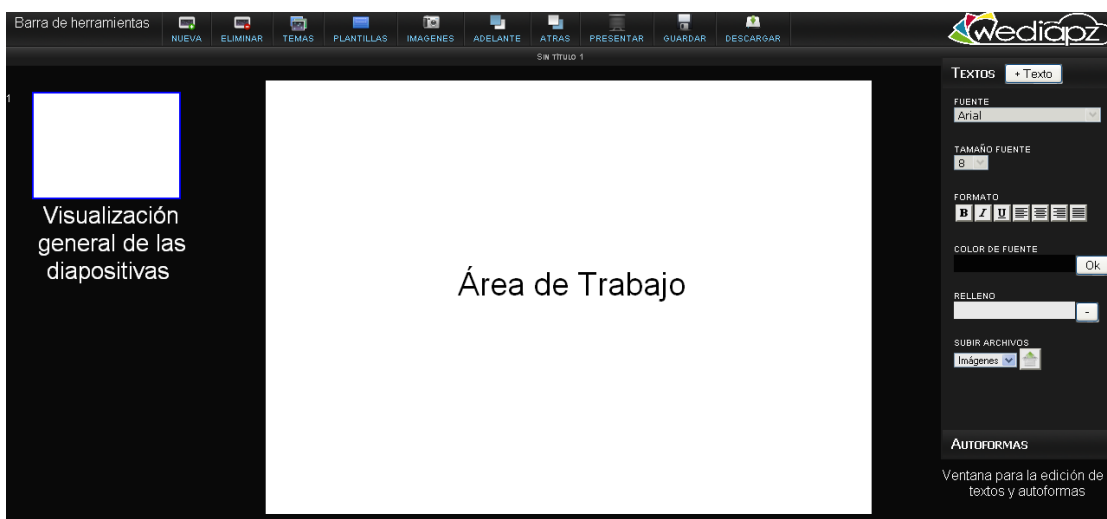
**Figura 4.6 Interfaz página principal. [20]**



### 4.5.3 Editor Wediapz

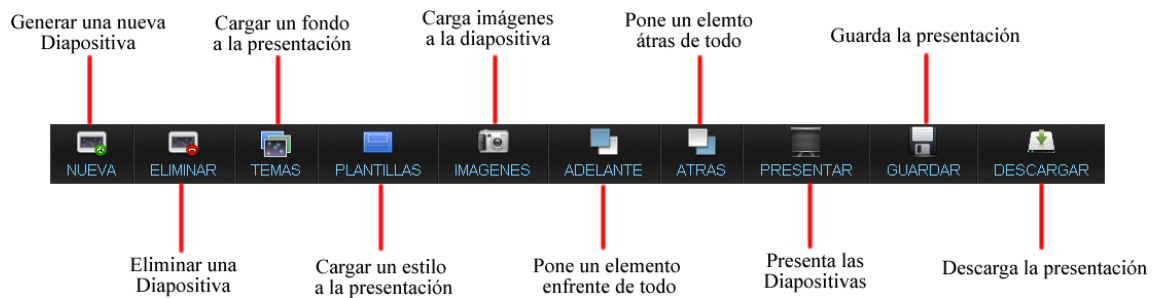
El editor Wediapz es la herramienta con la cual se crean y editan las presentaciones el editor está compuesto por cuatro secciones mostradas en la figura 4.7, estas son el área de trabajo, la visualización general de las diapositivas las funciones generales, las funciones de estilo de texto, subir archivos y autoformas.

**Figura 4.7 Interfaz del editor Wediapz. [20]**



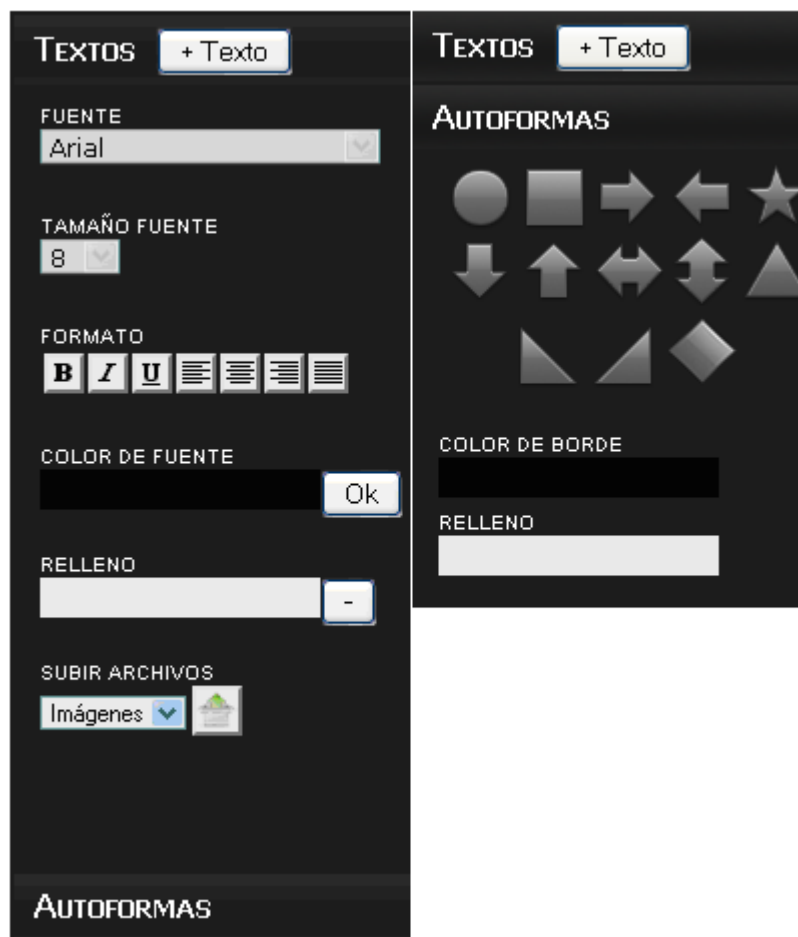
En la figura 4.8 observamos las funciones que podemos realizar al editar una presentación.

**Figura 4.8 Funciones generales del editor. [20]**



En la figura 4.9 vemos las funciones para el estilo de texto, opciones como centrar letra, tipo de letra, tamaño, color, al igual que la función de subir archivos ya sean imágenes o temas. También esta la opción de las autoformas.

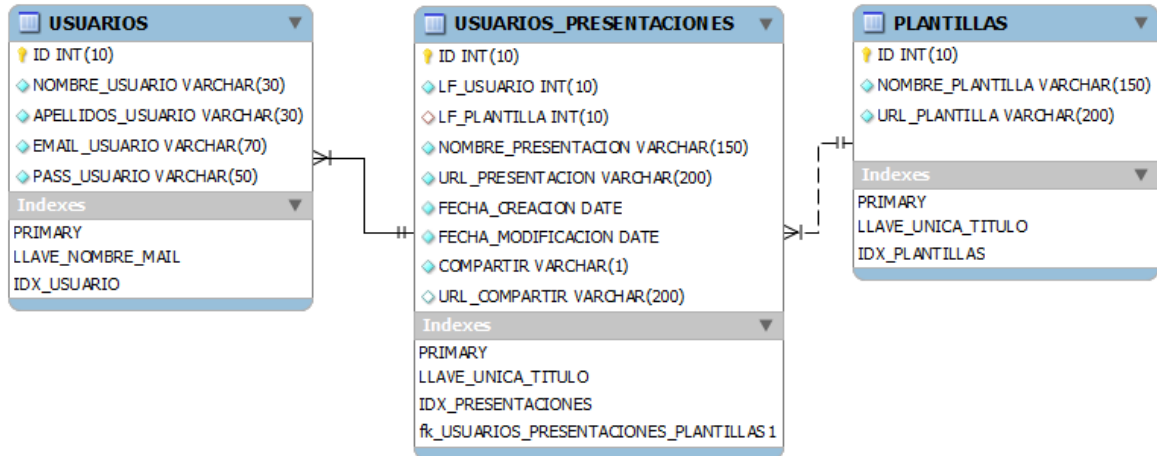
**Figura 4.9 Funciones para el estilo de texto. [20]**



## 4.6 Base de Datos

En este diagrama se muestra la base de datos del sistema y las relaciones que tienen una tabla con otra.

**Diagrama 4.d Diagrama entidad – relación de la base de datos Wediapz. [20]**



En el diagrama entidad-relación de la base de datos, la entidad usuario tiene como atributos:

- ID
- Nombre de usuario
- Apellido de usuario
- E-mail de usuario
- Password

Tiene como llave primada el ID y el email sirve como un atributo único, la entidad usuario se relaciona 1:N con la entidad usuarios\_presentaciones, la cual tiene como atributos:

- ID
- Nombre de presentación
- URL de presentación
- Fecha de creación
- Fecha de modificación
- Compartir
- URL compartir

La entidad usuarios\_presentaciones tiene dos atributos más que son llaves foráneas de la entidad usuarios y la entidad plantillas, la entidad usuarios\_presentaciones tiene como llave primaria el ID y un atributo único que es el nombre de la presentación se relaciona con N:1 con la entidad usuarios y N:N con la entidad plantillas.

La entidad plantillas es para compartir las presentaciones con otros usuarios cuenta con estos atributos:

- ID
- Nombre de plantilla
- URL plantilla

Su llave primaria es el ID y al igual que las demás entidades tiene un atributo único que es el nombre de la plantilla.

#### **4.7 Estructura general de directorios del sistema**

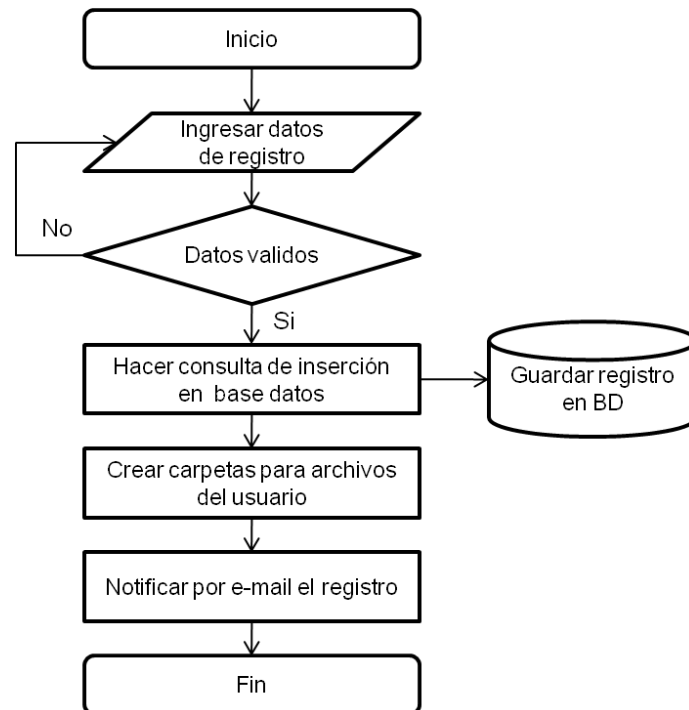
El sistema cuenta con un directorio específico para alojar las presentaciones creadas por el usuario así como las imágenes y temas que puede subir para su posterior uso en las presentaciones, en el caso de las dos últimas y el de las presentaciones para descargarla o compartirla. En el esquema 4.e se muestra como están estructurados dichos directorios.

##### **Esquema 4.e Esquema de directorios de wediapz. [20]**

- archivos
  - idusuario 1
    - imágenes
    - presentaciones
    - temas
  - idusuario 2
    - imágenes
    - presentaciones
    - temas
  - 
  - 
  - 
  - idusuario n
    - imágenes
    - presentaciones
    - temas
  - temp
  - compartir
    - presentaciones
    - temas

El directorio de cada usuario es creado cuando se registra en el sistema y está representado por el diagrama de flujo 4.f que explica el funcionamiento básico.

**Diagrama 4.f Diagrama de Flujo para registro de un usuario. [20]**



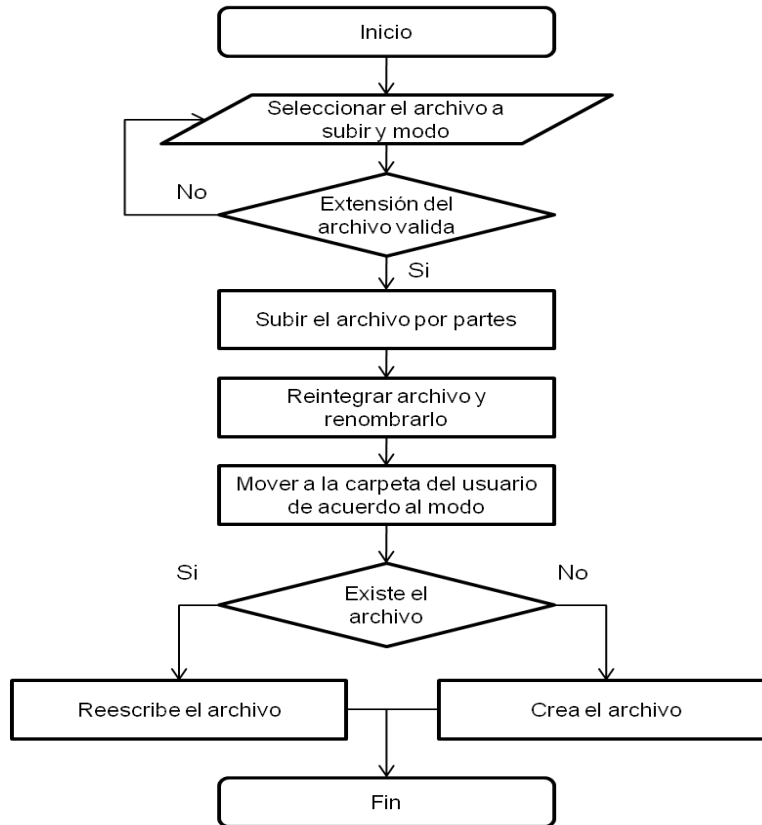
Para subir archivos a los directorios existen tres modos dentro del sistema que sirven para identificar que extensiones son validas en cada modo así como para ubicar los archivos en sus correspondientes carpetas, a continuación se mencionan dichos modos y sus extensiones validas.

- **Modo Imágenes:**
  - Coloca los archivos en la carpeta de imágenes del usuario.
  - Extensiones validas (jpg, jpge y png).
  
- **Modo Temas:**
  - Coloca los archivos en la carpeta de temas del usuario.
  - Ajusta las imágenes a un tamaño defino de 700px por 500px.
  - Extensiones validas (jpg, jpge y png).
  
- **Modo Presentaciones:**
  - Coloca los archivos en la carpeta de presentaciones del usuario.
  - Registra en la base de datos las presentaciones del usuario.
  - Extensiones validas (jpg, jpge y png).



En el diagrama de flujo 4.g se muestra la generalización del procedimiento de subir archivos.

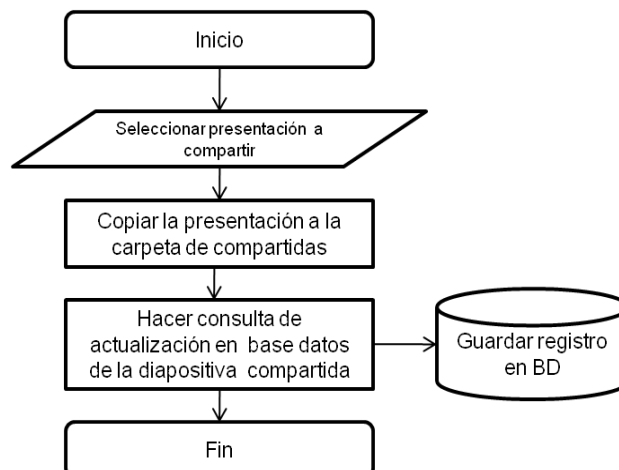
**Diagrama 4.g Diagrama de Flujo para subir archivos. [20]**



#### 4.8 Compartir presentaciones

Las presentaciones compartidas son presentaciones creadas por los usuarios y puestas en la carpeta de compartir, estas presentaciones están disponibles para todos los usuarios y únicamente pueden ser descargadas por los demás usuarios, en el diagrama de flujo 4.h se muestra el procedimiento para compartir una presentación.

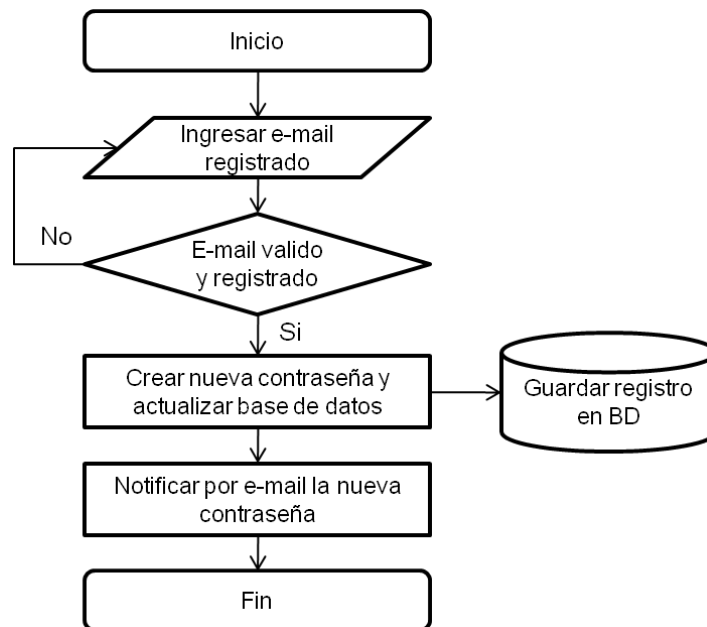
**Diagrama 4.h Diagrama de Flujo para compartir presentaciones. [20]**



## 4.9 Recuperar contraseña

Cuando un usuario olvida su contraseña el sistema es capaz de generar una nueva contraseña para ese usuario y a su vez actualizar la base de datos con la nueva contraseña y mandársela por e-mail al usuario, para llevar a cabo este procedimiento es necesario estar registrado en el sistema y contar con un e-mail valido. En el diagrama de flujo 4.i se representa este procedimiento.

**Diagrama 4.i Diagrama de flujo para recuperar contraseña. [20]**



## CAPITULO V. Pruebas

El enfoque de pruebas será el mismo que se haya recomendado para las pruebas de sistemas orientados a objetos, este se resume:

- El modelo de contenido es revisado para describir errores
- El modelo de diseño es revisado para descubrir errores de navegación.
- Se aplican pruebas de unidad a los componentes de proceso seleccionados y las páginas Web
- La aplicación Web ensamblada se prueba para conseguir una funcionalidad global.
- La aplicación Web se implementa en una variedad de configuraciones diferentes de entornos y comprobar así la compatibilidad con cada configuración.

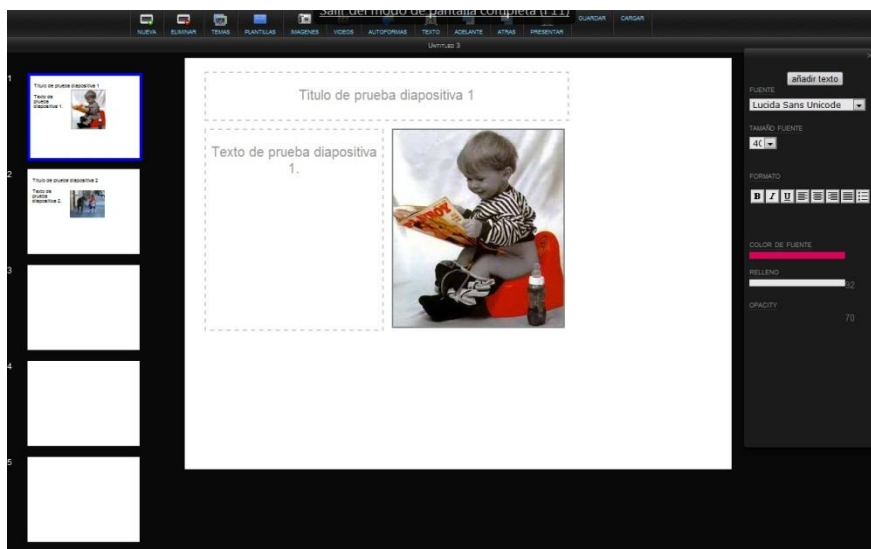
La aplicación Web se prueba con una población de usuarios finales controlada y monitorizada

### 5.1 Uso del editor Wediapz

El editor es capaz de leer un archivo en ODP y editarlo, sus funciones consisten en agregar plantillas de texto, temas para el fondo de la presentación, así como modificar los textos de las diapositivas, agregar imágenes, añadir y eliminar diapositivas y guardar las diapositivas.

Un ejemplo de esto son las siguientes figuras que muestran la funcionalidad del editor.

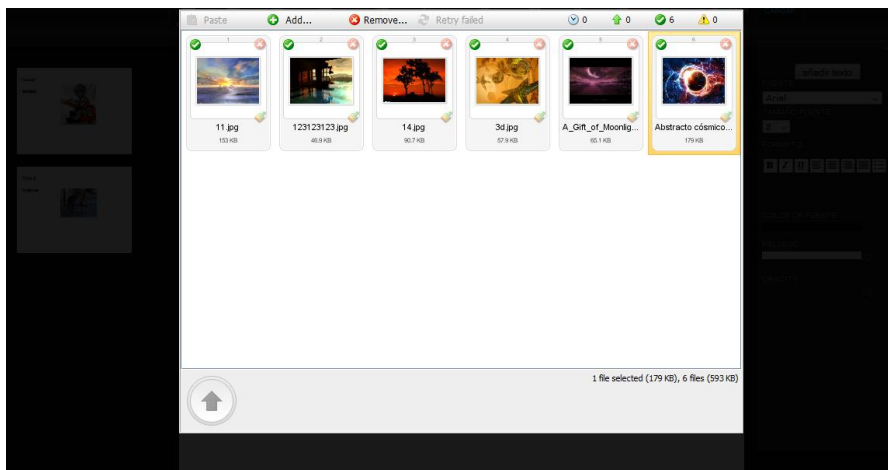
**Figura 5.a Funciones del editor Wediapz (antes). [20]**



**Figura 5.b Funciones del editor Wediapz (después). [20]**



**Figura 5.c Funciones del editor Wediapz (subir imágenes). [20]**



Como se ve en la figura 5.a se ha cargado una presentación y se le han añadido más diapositivas, en la figura 5.b se han eliminado diapositivas y se le ha dado estilo a la primera diapositiva se le ha cambiado el relleno de un texto así como la fuente, el color y el relleno y con esto mostrando algunas de las funcionalidades del editor. En la figura 5.c se observa la función del editor de subir imágenes al servidor para poder añadirlas a su presentación.

## 5.2 Lectura del formato ODP

Para que el editor pueda cargar una presentación y pueda ser editada se desarrolló un archivo de JavaScript capaz de leer e interpretar un archivo en formato ODP en su forma descomprimida pues en apartados anteriores se vio que el ODP es un ZIP. Este archivo hace uso del Framework de JavaScript jQuery.

### **5.3 Crear nuevas diapositivas**

Para crear nuevas diapositivas en el editor se ha creado una función para crearlas y eliminarlas así como existen funciones que manipulan el estilo de las mismas, parte de estas funciones funcionan cuando un elemento esta seleccionado, estas funciones alteran el estilo de los elementos dependiendo de qué función se trate.

### **5.4 Guardar en formato ODP**

Para guardar en formato ODP usamos las librerías que ya trae incluidas java para guardar en este formato y el archivo es ejecutado desde PHP, este archivo lee un archivo de texto plano y le da formato para guardarlo, nosotros hemos creado este archivo y hemos creado nuestro formato para guardar y dar estilo a las presentaciones hechas en el sistema.

## **Conclusiones**

En el trabajo nos encontramos con algunas dificultades al implementar algunas funciones ya que no todos los navegadores funcionan de la misma manera e incluso no tienen implementado el uso del HTML versión 5, otra complicación fue el poder guardar las presentaciones en formato ODP ya que nosotros creamos un formato de guardar con los diferentes contenidos que posee una presentación.

La experiencia y los conocimientos que nos dejó el desarrollo de este proyecto fueron sobresalientes, ya que fue necesario reafirmar y aplicar muchos de los conceptos de programación Web, javascript, html, el uso de frameworks, diseño de interfaces gráficas de usuario altamente usables.

En vista de los resultados obtenidos en las pruebas finales podemos decir que el objetivo esencial de este Trabajo Terminal se cumplió, es decir, se pueden cargar y crear presentaciones para editarlas el poder compartir las presentaciones con los diferentes usuarios así como guardar las presentaciones en formato ODP.

## **Trabajo a futuro**

A mediano y largo plazo este trabajo podría extenderse para que desempeñara algunas funcionalidades

- Exportar las presentaciones a otros formatos PDF
- Agregar funciones como de insertar tabla, o hacer gráficos.
- Dibujar libremente sobre el área de trabajo.
- Crear una función para auto-corrección del texto.
- Introducir videos de mayor tamaño y diferentes formatos de video, al igual que de sonidos.

## Glosario

**API:** Una interfaz de programación de aplicaciones o API (Application Programming Interface) es el conjunto de funciones y procesamiento (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usados generalmente en las bibliotecas.

**CGI:** Interfaz de entrada común (Common Gateway Interface) es una importante tecnología de World Wide Web que permite a un cliente (navegador Web) solicitar datos de un programa ejecutado en un servidor Web. CGI especifica un estándar para transferir datos entre el cliente y el programa.

**Cookie:** Es un fragmento de información que se almacena en el disco duro del visitante de una página Web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas.

**DOM:** El Document Object Model (DOM), es esencialmente una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.

**JSON:** Acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

**Plug-in:** Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

**Servlets:** Son pequeños programas que se ejecutan en el contexto de un navegador Web. Por contraposición, un servlet es un programa que se ejecuta en un servidor. 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.

**Script:** Es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. Los script son casi siempre interpretados, pero no todo programa interpretado es considerado un script. El uso habitual de los scripts es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario.

**SQL:** Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar información de interés de una base de datos, así como también hacer cambios sobre ella.

**XML:** El XML (Extensible Markup Language) es un conjunto de reglas para codificar documentos de manera electrónica. XML es una simplificación y adaptación del SGML

(Standard Generalized Markup Language) y permite definir la gramática de lenguajes específicos.

**XMLHttpRequest (XHR):** Es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web. Para los datos transferidos se usa cualquier codificación basada en texto. La interfaz se presenta como una clase de la que una aplicación cliente puede generar tantas instancias como necesite para manejar el diálogo con el servidor. El uso más popular, si bien no el único, de esta interfaz es proporcionar contenido dinámico y actualizaciones asíncronas en páginas WEB mediante tecnologías construidas sobre ella como por ejemplo AJAX.

**Triggers:** Es un procedimiento que se ejecuta en forma inmediata cuando ocurre un evento especial. Estos eventos sólo pueden ser la inserción, actualización o eliminación de datos de una tabla.



## Referencias

- [1] Prezi. The zooming presentation editor, 18 de Septiembre de 2010, [en línea]. Available: <http://prezi.com>
- [2] 280Slides. Create and Share presentations online, 18 de Septiembre de 2010, [en línea]. Available: <http://280slides.com/>
- [3] Slide Rocket. Online presentation software, 18 de Septiembre de 2010, [en línea]. Available: [http://www.sliderocket.com/index\\_c.html](http://www.sliderocket.com/index_c.html)
- [4] Software online y Web 2.0, 21 de Julio de 2010, [en línea]. Available: <http://www.laflecha.net/articulos/softlibre/software-online-y-web-20>
- [5] Thomas A. Powell, “Manual de Referencia HTML4”, editorial Mc Graw Hill, Primera edición en español, 2001.
- [6] Filosofía Tecnológica, “Nuevos elementos en HTML5, la quinta versión del lenguaje Web”, 9 de Agosto de 2007, [en línea]. Available: <http://techlosofy.com/nuevos-elementos-en-html-5-la-quinta-version-del-lenguaje-de-la-web/>
- [7] Comisión Sectorial de Desarrollo Informático de la Universidad de la República, “Estándares de formato de archivos de ofimática”, Abril 2008. [en línea]. Available: [http://www.universidadur.edu.uy/prensa/noticias/images/imagenes\\_noticias/estandares\\_de\\_ofimatica\\_en\\_udelar\\_v61.pdf](http://www.universidadur.edu.uy/prensa/noticias/images/imagenes_noticias/estandares_de_ofimatica_en_udelar_v61.pdf)
- [8] José López Quijado, “Domine PHP5”, editorial Alfa Omega
- [9] PHP, “PHP Manual”, Noviembre 2010. [en línea]. Available: <http://www.php.net/>
- [10] Javier Eguíluz Pérez, “Introducción a JavaScript”, 4 de Mayo 2009, [en línea]. Available: <http://www.librosweb.es/javascript/index.html>
- [11] Danny Goodman, “JavaScript, Bible”, editorial Hungry Minds, gold edition, 2001.
- [12] Javier Eguíluz Pérez, “CSS Avanzado”, 18 de Junio 2009, [en línea]. Available: [http://www.librosweb.es/css\\_avanzado/](http://www.librosweb.es/css_avanzado/)
- [13] CSS3.info, “CSS3 Previews”, [en línea]. Available: <http://www.css3.info/>
- [14] Javier Eguíluz Pérez, “Introducción a AJAX”, 31 de Agosto 2009, [en línea]. Available: <http://www.librosweb.es/ajax/index.html>
- [15] Aplicaciones Web Serv Inf UA, “Curso librerías Web 2.0”, Noviembre 2008, [en línea]. Available: [http://www.ua.es/es/servicios/si/documentacion/mootools/introduccion/t0\\_introd.pdf](http://www.ua.es/es/servicios/si/documentacion/mootools/introduccion/t0_introd.pdf)
- [16] Mootools, “Documentation”, [en línea]. Available: <http://mootools.net/docs/core>
- [17] jQuery, “Documentation”, [en línea]. Available: [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page)
- [18] H.G. y A. Silberschatz, “Fundamentos de BD”, Mc Graw Hill, Cuarta Edición
- [19] MySQL, “Why MySQL”, [en línea]. Available: <http://www.mysql.com/why-mysql/>
- [20] Personal