***.Índice***

Introducción

Capitulo I. Identificación y planteamiento del problema

I.1 Justificación

I.2 Objetivos

I.2.1 Objetivo general

I.2.2Objetivos específicos

I.3 Alcances y limitaciones

Capitulo II. Marco teórico

***Índice de figuras***

Figura 1.

***Introducción***

Hoy en día el uso de tecnologías web es cada vez más común y en muchos casos, necesario, para el desarrollo de aplicaciones, ya que el navegador web ha ganado terreno frente al software de escritorio, ya sea para mandar un correo, generar y/o compartir documentos, editar imágenes, reproducir vídeos, etc.

Es por esta razón que en este proyecto de investigación se documenta la integración de las tecnologías web actuales y de gran popularidad para desarrollar una aplicación para consultorios pequeños que necesiten gestionar las citas que se generan a diario y administrar la contabilidad de sus servicios y productos.

En muchos casos el diseñador de software trabaja bajo reloj, cualquier herramienta que le facilite el trabajo es bienvenida, es por ello que el uso de frameworks y librerías son cada vez más utilizadas por aquellos que necesitan entregar un sistema web eficiente, fácil de mantener, entendible y de calidad en poco tiempo.

Anteriormente era muy común que el lenguaje del lado del servidor y el lenguaje del lado del cliente estuvieran separados sin ninguna interacción entre sí, pero esto ha cambiado con la integración de tecnologías que sirven como puente de comunicación asíncrona entre el cliente y el servidor, haciendo la experiencia de navegar por la web más amigable y placentera.

Si el programador trabaja solo o en equipo, no es una mala idea el uso de un sistema de control de versiones ya que en muchos de los casos el primer camino que se elige no es siempre el correcto y es necesario regresar algunos o varios pasos atrás para rectificar, siendo este sistema una manera segura y rápida de mantener un respaldo de todo el trabajo y esfuerzo realizado.

***Capítulo I: Identificación y planteamiento del problema***

Al hacer el análisis de algunas clínicas particulares que ofrecen servicios como consultas médicas, análisis de laboratorio y venta de medicamentos, se detectan las siguientes problemáticas.

* No se cuenta con un sistema eficiente para gestionar las citas que se generan día con día.
* No se cuenta con un sistema que lleve a cabo la administración básica de la contabilidad.
* Las aplicaciones que se han implementado para resolver las problemáticas anteriores sean dejado de utilizar por lo complejo que resulta a los usuarios administrarlas ya que no tienen un conocimiento amplio en el uso de aplicaciones.
* No se cuenta con un solo software que administre las problemáticas anteriores y se tienen que utilizar dos diferentes sistemas.

Para dar solución a estas problemáticas, el presente proyecto integra las tecnologías web actuales para desarrollar un sistema de gestión de citas y administración de contabilidad.

**1.1. JUSTIFICACIÓN**

La mayoría de las empresas dedicadas al desarrollo de software, diseñan aplicaciones muy complejas, difíciles administrar y de utilizar para los usuarios que tienen conocimientos mínimos en el uso de ordenadores, es por esta razón que en el presente proyecto se desarrollara un software fácil de utilizar para la mayoría de las personas que tengan un conocimiento básico en el uso de aplicaciones.

Ya que la mayoría de los usuarios que tienen interacción con las computadoras, saben abrir y utilizar sin ningún problema un navegador web (Internet Explorer, Chrome, FireFox, Safari, etc.), ya sea para mandar correos y/o navegar por las redes sociales, es por esta razón que será más fácil que los usuarios se adapten a utilizar una aplicación que se ejecute en los navegadores, y es por esto que se decidido utilizar lenguajes de programación orientadas al desarrollo web.

Al implementar el sistema de gestión de citas en pequeños y medianos consultorios no solo contaran con una forma más fácil de administrar sus citas, también tendrán la opción de mandar y/o recibir correos electrónicos y mensajes de texto recordando al paciente y doctor la fecha y la hora de la próxima cita.

Con la implementación del módulo de contabilidad, podrán obtener reportes de compra/venta de artículos y servicios que se generen día con día para tener una mejor forma de saber cuánto se está gastando y cuánto dinero está generando el consultorio.

**1.2. OBJETIVOS**

**1.2.1. Objetivo General**

Desarrollo de un sistema web para consultorios médicos que gestione productos y servicios, administre citas y controle sus pacientes. El sistema debe de ser capaz de llevar el control de citas, mandar correos electrónicos y SMS como recordatorio de las próximas citas, debe llevar el control de los productos y servicios que ofrece el consultorio y mostrarlos sus gastos e ingresos de una manera clara, por último debe almacenar el historial de los pacientes para que puedan ser mostrados en cada cita.

**1.2.2. Objetivos Específicos**

* Desarrollar un modulo de gestión de usuarios con altas bajas y modificaciones al cual solo tendrá acceso el administrador del sistema.
* Desarrollar un módulo de gestión de empleados con altas, bajas y modificaciones.
* Desarrollar un módulo de gestión de pacientes e historial médico con altas, bajas y modificaciones.
* Desarrollar un módulo de servicios con altas, bajas y modificaciones.
* Desarrollar un módulo de productos con altas, bajas y modificaciones.
* Desarrollar un módulo que genere reportes según el lapso de tiempo que se requiera.
* Maquetación de todos los módulos con html5 y css3.

**1.3 Alcances y limitaciones**

El sistema está pensado para que se ejecute en navegadores web y los pacientes que cuenten con acceso a internet podrán podrán recibir correos electrónicos recordándole la fecha de su próxima cita, es por esta razón que los consultorios cuenten con una conexión continua a la nube, en caso de que los consultorios no tengan acceso a esta servicio se podrá instalar una máquina que sirva como servidor local sin la posibilidad de que los paciente puedan recibir correos electrónicos.

El envío de SMS también está restringido por estas limitaciones y al ser un servicio que se contrata con una empresa dedicada a este rubro, el envío de mensajes de texto también está restringido por la cantidad de SMS que se contraten con dicha empresa.

En el caso de la contabilidad del consultorio es muy importante que el administrador del sistema tenga al día la información de sus ventas y gastos que se generen a diario para que el sistema arroje reportes con información actualizada y real.

Al diseñar la plantilla del sistema con tecnologías actuales los exploradores más antiguos no podrán soportar al cien por ciento las características de la plantilla.

***Capítulo II: Marco teórico***

A continuación se explican algunos conceptos básicos necesarios para un mejor entendimiento del desarrollo de este proyecto, así como una breve historia de la web, para mostrar la importancia que se ha tenido en la comunicación mundial y que se espera de la web para un futuro no muy lejano.

**¿Qué es Web?**

La World Wide Web (WWW) o más popularmente conocido como “la web” es una red informática mundial y no es mas que un sistema de distribución de documentos que contienen hipertexto que son interpretado por navegadores web; estos navegadores muestran el hipertexto al usuario de una manera entendible y con un mejor aspecto.

Los documentos de hipertexto están almacenados en computadoras llamadas servidores, que son los encargados proveer estos documentos a los clientes que lo soliciten, estas peticiones se hacen por medio de una dirección IP (ej. 173.194.46.31) que es traducida en una URL (ej. [www.google.com.mx](http://www.google.com.mx/)) por la base de datos distribuida de internet conocida como DNS.

**Historia de la web**

La primer comunicación web entre un servidor y un cliente fue en noviembre de 1989 y la realizo Tim Berners-Lee en las oficinas de CERN. La primer página HTML original de perdió, pero en la siguiente liga se encuentra una copia de esta primer página web: [www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html](http://www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html).

Tim Berners-Lee es considerado como *el padre de la web,* la razón de este sobrenombre es por que desarrollo las tres principales tecnologías para la comunicación web: HTML siglas de **HiperText Markup Language** (lenguage de marcas de hipertexto), URL siglas de **Uniform Resource Locator** (localizador de recursos uniformes), HTTP siglas de **Hipertext Transfer Protocol** (protocolo de transferencia de hipertecto).

El principal objetivo de Tim Berners-Lee era **compartir** la información que generaban los científicos del mayor laboratorio de investigación en física de partículas a nivel mundial (CERN). El problema de dichos científicos era no poder dar a conocer el resultado de sus investigaciones de un manera rápida y eficiente, para resolver esta problemática Tim creo un formato para escribir los documentos (HTML):



Figura 1 Ejemplo básico de HTML

Desarrollo un método para representar los enlaces entre los documentos (URL):

<http://www.youtube.com/>

Y definió un protocolo de red para transmitir los documentos de un ordenador a otro (HTTP)

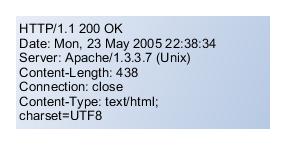


Figura 2 Ejemplo del protocolo HTTP

Ejemplo de la visualización de un archivo html en un navegador web:

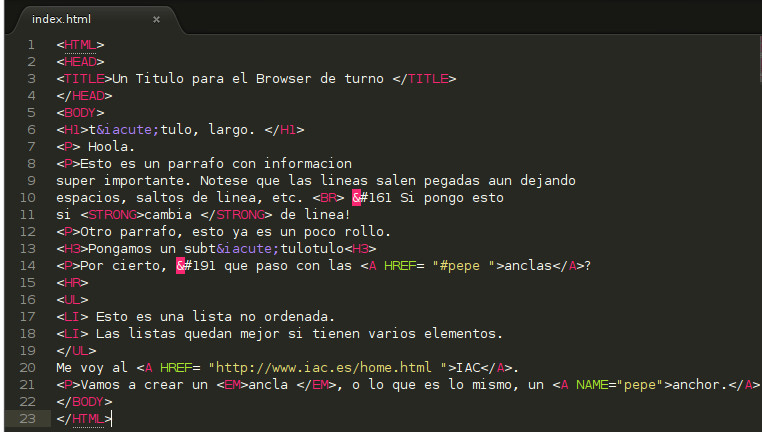
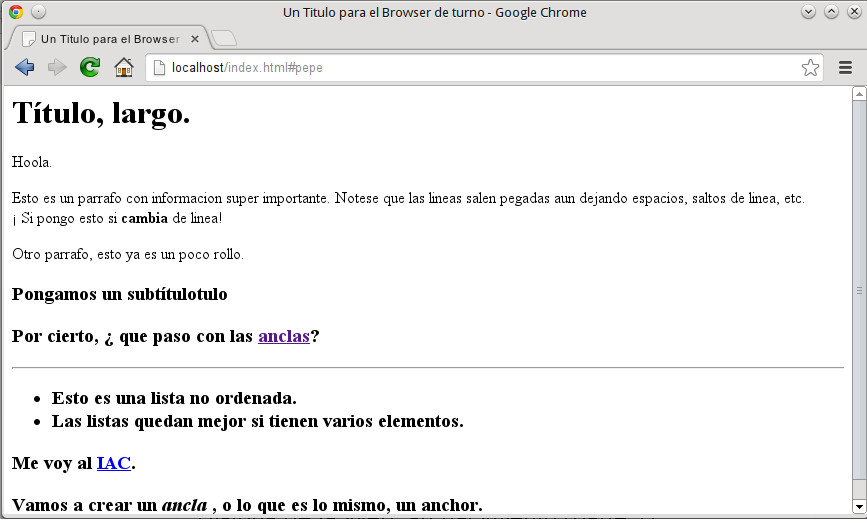


Figura 3 Ejemplo de la visualización de un archivo html en un navegador web

Ref. 1 <http://www.iac.es/galeria/westend/ejemplo.html>

**Evolución de la web**

Ahora veremos el crecimiento progresivo que ha tenido la web hasta llegar a lo que ahora conocemos como web 3.0, haremos un recorrido por las diferentes etapas y se enlistan las características de cada una de ellas.

**WEB 1.0 (WEB Primitiva)**

Es la primera etapa de la web. En esta etapa las paginas web solo contenían texto que los usuarios únicamente podía visualizar sin ninguna interacción con el contenido de la misma. Aparecen los primeros navegadores web como Netscape e Internet Explores en sus primeras versiones, estos navegadores son bastantes rápidos ya que solo se visualizaba texto.

**Características:**

* Pocos productores de contenido.
* Muchos lectores de estos contenidos.
* Páginas estáticas.
* Las actualizaciones de las paginas no se realiza de forma periódica.
* Interacción mínima reducida a formularios de contacto, inscripción de boletines, etc.

**WEB 2.0 (WEB Social)**

La WEB 2.0.es la transformación que han tenido las aplicaciones que funcionan a través de la web, las aplicaciones pasaron de mostrar solo contenido que los editores publicaban a aplicaciones cuya información la generan los mismo usuarios que las usan, también se han desarrollado servicios que intentan sustituir las aplicaciones de escritorio.

Hay que poner en claro que la web 2.0 no es una tecnología como tal o la versión de alguna paquetería de software, se trata de como se utilizan las tecnologías de forma diferente, ya sea para dar un mejor aspecto o ser mas amigables con el usuario final, pero la web 2.0 no se podría haber dado sin la ayuda de diferentes tecnologías e implementaciones como son:

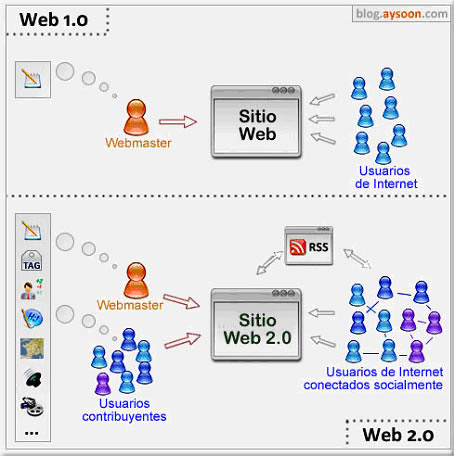
* **CSS:** permitió la separación del código html y los estilos en diferentes archivos
* **AJAX:** es la suma de varias tecnologías que permite hacer peticiones al servidor de manera asincrónica.
* **Flash:** Animaciones e interacción con los usuarios.
* **XML** para el intercambio de contenido.
* Migraciones de software de escritorio a hacia la plataforma web.
* Sindicación de contenidos (XML).
* Redes Sociales alimentadas por los usuarios.
* API's para la manipulación de aplicaciones

**Características:**

La característica principal de la web 2.0 es la participación de los usuarios al compartir información continuamente y no solo como espectador de los contenidos de la web.

* Aparición de redes sociales.
* Aparición de blogs.
* Aparición de wiki's.
* La implementación de CMS para la creación de páginas web.
* Aplicaciones web dinámicas.

**Comparación WEB 1.0 y WEB 2.0**

Figura 4 Comparación WEB 1.0 y WEB 2.0 

Ref. 2 http://www.eduteka.org/Web20Intro.php

**WEB 3 (WEB Semántica)**

Actualmente se pueden encontrar diferentes conceptos respecto a esta etapa de la WEB y es complicado encontrar la definición más adecuada, pero en lo que concuerdan la mayoría de las explicaciones sobre esta web, es que la web debe ser mas intuitiva, mejor interrelación entre las aplicaciones, facilidad de acceso y lo mas importante la integración de la inteligencia artificial.

Ciertamente aun no se ha llegado por completo a esta etapa pero solo es cuestión de tiempo para alcanzar la, ya que muchas de las tecnologías requeridas para hacer posible la WEB 3.0 poco a poco se están poniendo en marcha. A continuación se enlistan estas tecnologías

* RDF: para los metadatos.
* OWL: es un lenguaje de marcado para compartir datos usando ontologías.
* JENA: framework de java.
* SPARQL: Estándar para la consulta de RDF

**WEB 4.0 (WEB Omnipresente)**

Si bien es cierto que aun no hemos al cazado la etapa anterior, ya se esta pensando en lo que la web puede llegar hacer en un futuro y la meta de esta etapa es poder tener información y comunicación en cualquier lugar, en donde nos encontremos.

Microsoft Office nos muestra un vídeo de como podría llegar hacer el mundo en un futuro y en el cual se enfatiza el papel fundamental que tiene la web

<https://www.youtube.com/watch?v=MxriE6g3G9c>

**Tecnologías usadas en el proyecto**

Ya se ha presentado un poco de la historia de la plataforma en la que se ha decidido desarrollar el presente proyecto. Ahora toca el turno de definir las tecnologías web que se implementan para la ejecución del software para la administración de citas y contabilidad de consultorios médicos.

**Del lado del servidor**

**PHP (Framework Codeigniter)**

PHP: Hypertext Preposesor, es un lenguaje de programación interpretado ejecutado del lado del servidor, este lenguaje fue desarrollado originalmente para la creación de páginas web dinámicas, fácil de aprender ya que se puede programar de manera estructurada u orientada a objetos, es embebido fácilmente en código HTML, conexión fácil y eficiente con bases de datos; por estas razones es uno de los lenguajes más utilizados por desarrolladores de aplicaciones web.

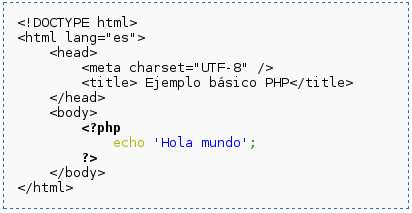


Figura 5 Programa “Hola Mundo”con PHP embebido en código HTML:

Ref. 3 http://es.wikipedia.org/wiki/PHP

Para los desarrolladores que tienen conocimiento de otros lenguajes de programación como C, Java, C#, Pascal etc. La salida de los resultados normalmente en la pantalla , en PHP la salida es una página HTML que posteriormente interpreta el navegador web.

**Historial de lanzamiento**

|  |  |
| --- | --- |
|  | Significado |
| Rojo | Versión antigua, sin soporte |
| Amarillo | Con soporte para errores de seguridad |
| Verde | Con soporte para errores generales y de seguridad |
| Azul | Lanzamiento futuro |

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha de lanzamiento | Fin de soporte | Notas |
| 1.0 | 08-06-1995 |  | Llamada oficialmente "Personal Home Page Tools (PHP Tools)". Es la primera versión en usar el nombre "PHP". |
| 2.0 | 01-11-1997 |  | Considerada por sus creadores la "herramienta más rápida y sencilla" para crear páginas web dinámicas. |
| 3.0 | 06-06-1998 | 20-10-2000 | Zeev Suraski y Andi Gutmans reescribieron la base para esta versión. |
| 4.0 | 22-05-2000 | 23-01-2001 | Se añadió un sistema de parsing de dos fases llamado motor Zend. |
| 4.1 | 10-12-2001 | 12-03-2002 | Se introducen 'superglobales' ($\_GET, $\_POST, $\_SESSION, etc.) |
| 4.2 | 22-04-2002 | 06-09-2002 | Se deshabilitan register\_globals por defecto. Datos recibidos por la red no son insertados en el espacio de nombres global, cerrando posibles agujeros de seguridad en las aplicaciones. |
| 4.3 | 27-12-2002 | 31-03-2005 | Se introduce CLI, y CGI. |
| 4.4 | 11-07-2005 | 07-08-2008 | Se añaden páginas man para phpize y php-config. |
| 5.0 | 13-07-2004 | 05-09-2005 | Motor Zend II con un nuevo modelo de objetos. |
| 5.1 | 24-11-2005 | 24-08-2006 | Mejoras de rendimiento con la introducción de variables de compilador para el nuevo motor de PHP. |
| 5.2 | 02-11-2006 | 06-01-2011 | Activida extensión de filtro por defecto. Soporte JSON nativo. |
| 5.3 | 30-06-2009 | 07-2014 | Soporte para espacios de nombres, enlace estático en tiempo de ejecución, etiqueta de salto (goto limitada), Clausuras nativas, Soporte nativo para archivos PHP (phar), recolección de basura para referencias circulares, soporte mejorado para Windows, sqlite3, mysqlnd como reemplazo a libmysql como biblioteca para extensiones que funcionan con MSQL, fileinfo como reemplzado de mime\_magic para mejor soporte MIME, extensión de Internacionalización, y etiqueta ereg obsoleta. |
| 5.4 | 01-03-2012 | No especificada | Soporte para Trait y sintaxis abreviada de array. Elementos removidos: register\_globals, safe\_mode, allow\_call\_time\_pass\_reference,session\_register(), session\_unregister() y session\_is\_registered(). Servidor web incorporado. Varias mejoras a características existentes y al rendimiento, y requerimientos de memoria menores. |
| **5.5** | 20-06-2013 | No especificada | Nuevos generadores para bucles, empty() soporta expresiones. Se pierde el soporte para Windows XP y Windows Server 2003. |
| 6.0 | Sin fecha | No especificada | El desarrollo de PHP 6 ha sido retrasado porque los desarrolladores decidieron que el enfoque actual para tratar cadenas Unicode no es correcto, y están considerando formas alternas para la siguiente versión. Las mejoras planeadas para PHP 6 fueron añadidas en su lugar en PHP 5.3.0 (Soporte para espacios de nombre, enlace estático en tiempo de ejecución, funciones lambda, clausuras, goto) y 5.4.0 (traits, revinculación de clausura). |

Tabla 1 Historia de lanzamiento

Ref. 4 [http://es.wikipedia.org/wiki/PHP#cite\_note-about\_PHP](http://es.wikipedia.org/wiki/PHP" \l "cite_note-about_PHP)

**PHP Orientado a Objetos (POO)**

A partir de la versión 5 de php, se mejoró el soporte para la programación orientada a objetos.

La programación orientada a objetos es una forma distinta de programar y se asemeja más a como pensamos las cosas en la vida real. La POO intenta simular el mundo, con el uso de objetos que están formados por características y funciones que conforman una entidad de programación. Los objetos deben colaborar entre ellos para realizar alguna acción, esto nos permite tener una código mas limpio, fácil de entender, poco tiempo de mantenimiento y reutilizar código.

Para empezar a programar orientado a objetos es importante pensar igual que como lo haríamos en la realidad, como por ejemplo en la construcción de una casa; la casa es el elemento principal (Objeto), el plano de la casa (Clase), color, tamaño, número de habitaciones (Atributos) y acciones como el abrir y cerrar puertas o ventanas (Funciones o Métodos).

**Características de la Programación Orientada a objetos.**

Para que un leguaje de programación pueda ser orientado a objetos, es cuando reúnen las características de **abstracción, encapsulamiento, herencia y polimorfismo**; y los conceptos que lo forman: **objetos, mensajes, clases, instancias y métodos.** Para dejar mas claro de que se esta hablando se definirar los conceptos básicos de la POO:

**Abstracción:** Son las características particulares de cada objeto, las cuales lo diferencia de los demás objetos. Por ejemplo si aplicamos la abstracción a un automóvil. El objeto seria vehículo y sus atributos serian: color, tipo, modelo, marca, etc. y las funciones serian: acelerar, frenar, encender, apagar, etc.

**Encapsulamiento:** Esta característica se logra gracias a la abstracción y consiste en condensar en la clase las variables y métodos para mantenerlos ocultos y que estos no puedan ser manipulados por otros objetos.

**Polimorfismo:** Es la forma en que los objetos de una clase se comportan al mismo evento dependiendo de los parametros que se utilizan al momento de invocarlos.

**Herencia:** Es la propiedad que permite a los objetos ser creados a partir de otros ya existentes, obteniendo los métodos y atributos ya existentes en el objeto

**HTML 5**

**CSS3**

**Base de datos (MySQL)**

**Servidor web (Apache)**

**Del lado del cliente**

**JavaScritp (Libreria jQuery)**

**Navegador web**

**Comunicación Asincrónica**

**Web Service (AJAX)**

**Navegador Web**

**Metodología (SCRUM)**

**Repositorio (Git)**