**Convenciones:**

* **Las referencias van entre corchetes y en cursiva dentro de un recuadro**
* **El código fuente debe ir formateado y coloreado y con fuente monoespaciada**
* **Notas aparecen rodeada de dos líneas con letras azules**
* **Todas las clases tienen nombres con la primera en mayúsculas**
* **El nombre de las funciones o métodos siguen la regla de estilo camelCase a excepción los de los controladores**

**Temario.**

**1. Capítulo I**

**1.1. Definición Y Origen De Internet**

Antes de empezar a desarrollar el trabajo voy a citar la definición que Wikipedia da a la palabra Internet:

*[Internet es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial….]*

En conclusión podríamos decir que todos los equipos que tiene acceso a Internet son parte de él, la forma que tiene de comunicarse con otros dispositivos en esta gran red es a través de protocolos el más usado es el WWW, también existe otro protocolo el cual es el más usado en la actualidad y es el HTTP, ambos protocolos realizar transferencia de HIPERTEXTO entre dos equipos o dispositivos, veamos las definiciones de los términos más importantes cuando se habla de Internet:

**SERVIDOR**

Equipo encargado de atender peticiones web y responderlas enviando Información a su solicitante a través de protocolos de comunicación.

**CLIENTE**

Equipo o Dispositivo con capacidad de interactuar con un servidor, enviando y recibiendo información con la ayuda de protocolos de comunicación, esta interacción es posible gracias a los navegadores web.

**NAVEGADOR WEB**

Es una aplicación que funciona en el dispositivo del cliente, como dice su nombre es un navegador, utiliza la red de Internet para encontrar contenidos y comunicarse con equipos remotos. La funcionalidad básica del navegador es permitir la visualización de contenidos.

**WWW**

Es una serie de protocolos que permiten el envió de información de un equipo a otro, fue el primer estándar creado para la transmisión de datos entre equipos, se necesita de un navegador web para que este muestre los contenidos de forma gráfica.

**HTTP**

Mi intención no es el de dar nuevas definiciones a las cosas, sino de hacer entender los términos, por lo cual voy a citar nuevamente a Wikipedia quien tiene un concepto muy claro de lo que es HTTP.

*[Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador web o un spider) se lo conoce como "user agent" (agente del usuario). A la información transmitida se la llama recurso y se la identifica mediante un localizador uniforme de recursos (URL). Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.*

*HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener estado. Para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones web instituir la noción de "sesión", y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado…]*

**HIPERTEXTO**

Es el texto que podemos ver en la pantalla de cualquier dispositivo electrónico. Un hipertexto consta de los siguientes elementos: nodos o secciones, enlaces o hipervínculos y anclajes.

Los nodos son las partes del hipertexto que contienen información accesible para el usuario. Los enlaces son las uniones o vínculos que se establecen entre nodos y facilitan la lectura secuencial o no secuencial por los nodos del documento. Los anclajes son los puntos de activación de los enlaces.

* + 1. **Orígenes**

EL origen de Internet se remota a los años 1960, cuando los computadores estaban aun en su “infancia” los mismos que no podían comunicarse con otros equipos ni se podía trabajar de manera colaborativa dentro de una organización, vista esta necesidad nace la primera red llamada ARPAnet (Advanced Research Projects Agency Network o Red de la Agencia para los Proyectos de Investigación Avanzada de los Estados Unidos), que nos legó el trazado de una red inicial de comunicaciones de alta velocidad a la cual fueron integrándose otras instituciones gubernamentales y redes académicas durante los años 70[1].

*Nota: Recomiendo ver el siguiente video… http://www.youtube.com/watch?v=i4RE6dBAjH4*

* + 1. **Internet En La Actualidad**

Internet es hoy una necesidad más que un lujo, es una herramienta que nos ayuda a realizar todas las tareas ya sean productivas o de ocio, su difusión es tan grande que se lo usa para todo, esto es porque estamos en una era donde las comunicaciones son lo más importante, y el internet es uno de los medios más grandes y el más usado para llevarlas a acabo, si pensamos en las actividades que realizamos a diario estoy seguro que en la mayoría el internet está presente, ya sea para estudiar, para trabajar, pero si es una persona que no trabaja en una oficina o simplemente no usa un computador no significa que no use internet, el uso de internet no es solamente a través del computador o celular, cuantas veces vamos al banco o mejor aun quien no ha retirado dinero de un cajero electrónico, este dispositivo es un terminal o un cliente del servidor del banco y está conectado a través de internet.

Veamos otro ejemplo, para este ejemplo vamos a pensar en el gobierno antes todos los tramites se los realizaba de manera presencial en la institución, ahora en la mayoría de ellas muchos de sus trámites se los realiza por internet.

En realidad hay muchas cosas que llegan a nosotros a través del Internet y sin la necesidad de usar un computador. Lo que intento decir es que el Internet es un recurso al que le debemos dar su importancia y usarlo para nuestro beneficio, sacando el mayor provecho de él.

* 1. **Definición De Sitios Web, Su Estructura E Importancia**

Un sitio web es un conjunto de páginas web alojadas en un servidor web, por lo general las páginas del sitio tienen contenidos relacionados a un tema en particular o a un propósito, los contenidos están conectados a través de hiperenlaces que son los que guían al navegador para mostrarlos al usuario, a menudo se confunde sitio web con pagina web, la pagina no es más que un archivo con código HTML el cual es interpretado por el navegador y mostrado en forma clara al usuario, para dejar más claro el concepto veamos un ejemplo:

[www.google.com](http://www.google.com) es un sitio web el cual contiene una gran cantidad de páginas, si ingresamos al sitio nos muestra la página principal que es un formulario para introducir un texto, el cual nos ayuda a realizar búsquedas de algo en Internet.

Un sitio tiene los siguientes elementos:

* Dirección IP única, la de Google.com es 74.125.130.99
* Nombre de Dominio asignado a la IP www.google.com (dirección web)
* Estructura de árbol o jerárquica: A partir de una página de bienvenida o portal (raiz) se abren unas secciones (ramas) que a su vez contienen múltiples páginas web (hojas)
* Estructura lineal: a partir de una página de bienvenida o portal se suceden las siguientes páginas una tras otra como si se tratara de un libro.
* Estructura en red: Las páginas que forman el sitio web se enlazan unas con otras según sus contenidos en una especie de red en la que no se aprecia ningún tipo de jerarquía.

Existen dos tipos de Sitios Web:

**Sitios Estáticos**

Son sitios incapaces de interactuar con el usuario, sus contenidos se mantienen a menos que una persona con conocimientos de HTML los cambie manualmente, podemos decir que no tienen implementado un gestor de contenidos, en la actualidad existen muy pocos de estos sitios, aunque es frecuente encontrarlos en sitios web de empresas pequeñas, esto debido al costo que tiene implementar un sitio dinámico.

**Sitios Dinámicos**

Estos sitios aparecen gracias a la creación de programas informáticos desarrollados para el manejo de información del lado del servidor, antes del aparecimiento de estos lenguajes de programación, las paginas del sitio tenían que ser generadas manualmente por un webmaster, pero ahora gracias a la creación de los lenguajes de programación del lado del servidor, es posible crear una sola plantilla para mostrar los datos, la misma que esta basada en HTML o XHTML dependiendo las necesidades del sitio, almacenar los datos en un sistema gestor de bases de datos o incluso en ficheros de texto plano (sin formato), con esta tecnología ya no es necesario cambiar a una pagina web desde su estructura HTML sino que se puede cambiar directamente los contenidos a través del sistema de administración de sitio, mismo que es capaz de realizar las tareas mecánicas necesarias para mostrar la información al cliente, brindando al sitio la capacidad de interactuar con los usuarios, esto ha dado origen a un sinfín de aplicaciones como son blogs, foros, tiendas online, salas de chat, etc.

Podemos concluir diciendo que en un sistema dinámico, las paginas web no existen realmente porque son el resultado del procesamiento de la petición del usuario, el cual fue formado luego de que el servidor buscara la información y le diera forma para que el navegador del usuario la reciba, pero esa instancia de la pagina no existe es solo el resultado del algoritmo asociado a esa petición, esta es la principal diferencia entre sitios estáticos y dinámicos en los estáticos la pagina que estamos viendo si existe y es exactamente como el navegador nos la muestra.

**Importancia de los Sitios WEB**

La importancia de los sitios web está en lo que las empresas y personas pueden hacer con ellos, existes sitios en los que se pueden realizar comercio Mercado Libre es un gran ejemplo en Latinoamérica, buscadores como Google, Yahoo, Bing, etc. Tenemos también los servicios de correo electrónico, servicios de comunicaciones, redes sociales. Todos estos tienen un sitio en internet cuentan con un nombre único y una página principal que en la que vemos cuando nos conectamos al sitio.

* + 1. **Página Web**

Revisando la definición anterior podemos decir que página web es la unidad básica de un sitio web, es la encargada de mostrar los contenidos a los usuarios que la solicitan, estas no son mas que archivos en formato HTML o XHTML los cuales son interpretados por el navegador del cliente y luego mostrados de manera clara al usuario, la pagina web es la única forma en la que es usuario entra en contacto con el servidor.

* 1. **Introducción A los Frameworks**
     1. **Introducción**

Los Frameworks son ambientes de trabajo y ejecución con un conjunto de bibliotecas orientadas a la reutilización de componentes de software para el desarrollo rápido de aplicaciones, la palabra FrameWork traducida al español significa Marco de Trabajo, el principal objetivo de estas herramientas es ayudar al desarrollador a identificar los requerimientos y prestaciones del software, que tratando con los tediosos detalles de bajo nivel, sientan una base para poder comenzar a trabajar en lugar de crearla nosotros mismos, por ende la persona que los utiliza puede generar más con menos trabajo y en tiempos más cortos, podemos decir que un Framework es un esqueleto o esquema del sistema en donde nosotros vamos implementando lo que necesitemos.

Un Framework es más que una herramienta que permite desarrollar un proyecto fácilmente, es un conjunto de elementos que ayudan a nuestra aplicación a ser robusta y compatible, ya que estos manejan todo lo que tiene que ver con la seguridad y compatibilidad, las aplicaciones desarrolladas con un Framework no tienen problemas de compatibilidad de un sistema operativo a otro (dependiendo del Framework que se use), por ejemplo una aplicación realizada en .Net que es el Framework de desarrollo de Microsoft, funciona de manera idéntica en cualquier sistema operativo Windows siempre que tenga instalado el Framework. Veamos un ejemplo diferente, si se habla de una aplicación escrita en Java o Python las mismas pueden funcionar en más de un sistema operativo ya que el Framework es desarrollado para varios tipos de plataformas y equipos, lo que significa que una aplicación escrita es estos lenguajes puede funcionar de manera idéntica en Windows Linux Mac Unix, etc.

Continuando con lo bueno de estas herramientas, podemos hablar de un beneficio mas allá del código y del producto final, cuando un desarrollador o diseñador empieza a trabajar, toda la estructura esta a su cargo, lo que quiere decir que son ellos los encargados de generar la estructura básica del sistema, la pueden crear como mejor les parezca de acuerdo a su criterio profesional, la desventaja de esto es que la mayoría de estas personas sobre todo los principiantes no conocen de patrones de diseño y de técnicas que hacen que una estructura sea funcional y que pueda crecer a futuro, los Frameworks ayudan a estas personas a ser organizadas porque les ayuda a entender la estructura de un sistema, nos enseña a dividir un problema en pequeñas partes fáciles de manejar, esta para mi es una de las ventajas más importantes que tienen los Frameworks.

La estructura de los Frameworks no es general, no existe un modelo genérico para un Framework esto depende de la entidad encargada de su desarrollo, a continuación vamos a revisar una imagen de la estructura de Framework principal que se van a usar en este proyecto.

CodeIgniter creado por ellislab.com [2].



Como podemos observar los contenidos son mostrados y solicitados desde una página principal llamada index.php los mismos que pasan por filtros de seguridad y optimización antes de llegar al controlador de la aplicación que sería el corazón del Framework y esta a su vez usa a los componentes creados por el desarrollador como son los controladores librerías etc, para responder las peticiones del cliente.

Después de tanto hablar de lo bueno que tienen los Frameworks ha llegado la hora de hablar de los aspectos negativos que tienen.

Una de las principales desventajas, es que la aplicación desarrollada con la ayuda de un Framework es dependiente de este, ya que como dijimos anteriormente un Framework brinda componentes para facilitar el desarrollo, lo que quiere decir que la aplicación usa esos componentes, por lo tanto hay que escribirla casi por completo, mas de una persona puede decir. **¿Para qué cambiar de Framework?** La respuesta es sencilla, el Framework puede estar limitando la aplicación, puede ser que no nos deje desarrollar algún componente porque no fue diseñado para ello, aunque esto sucede rara vez, lo que si sucede es que el Framework consume recursos ya que controla la aplicación y la hace funcionar, para hacer que esto cambie el desarrollador puede generar la misma aplicación en un Framework que sea más optimo, la forma de evitar esto es elegir a un Framework con el que se pueda trabajar a diferentes escalas.

Otra desventaja es que el desarrollador no se vale por sí mismo para resolver algunos problemas sobre todo de compatibilidad, lo que intento decir es que más de una ventaja se convierte en desventaja para el crecimiento del desarrollador, si la vemos desde un punto de vista crítico.

* + 1. **Historia**

Creo que si algo está claro es la razón por la que fueron creados, por lo que no ahondaremos en el tema, lo único que voy a decir es que los Frameworks funcionan y es esa la razón por la que existen, a continuación voy a mostrar un grafico en el que se muestran la creación y evolución de algunos Frameworks a los largo del tiempo [3].



* + 1. **Tipos De Frameworks**

**Frameworks de Caja Blanca:**

La instanciación del Framework es posible a través de la creación de nuevas clases. Estas clases y el código correspondiente se pueden introducir por herencia o composición. Se agregan nuevas funcionalidades creando una subclase de una clase que ya existe en el Framework. Para usar Frameworks de caja blanca el desarrollador de aplicaciones debe conocer muy bien cómo funciona el Framework, este es el tipo de Framework que vamos a usar.

**Frameworks de Caja Negra:**

Producen instancias usando scripts de configuración del Framework, con los cuales se configura la aplicación final. Tienen la ventaja que no se requiere que el desarrollador de aplicaciones conozca los detalles internos del Framework, por lo cual son mucho más fáciles de usar.

**Frameworks de Caja Gris:**

La mayoría de los Frameworks son de Caja Gris, que son aquellos que contienen elementos de Caja Blanca y Caja Negra, ya algunas partes se implementan vía herencia o composición, y otras a través de configuración de parámetros [4].

**1.3.4. ¿Por qué usar un Framework?**

El porque esta en el motivo que impulso al ser humano a crearlos, gracias a la ayuda de estos podemos desarrollar aplicaciones de manera mas eficiente y rápida, mi intención no es sobrevalorar a los Frameworks, sino de mostrar su valor real a la hora de llevar a cabo el desarrollo de un proyecto, los Frameworks nos ahorran trabajo que mas que ser productivo es tedioso, que nos toma mucho tiempo y esfuerzo, voy a listar una serie de las tareas que hacen por nosotros.

* Manejan la seguridad de la aplicación
* Administran eficientemente una base de datos
* Desarrollo con poco código
* Mejores tiempos de desarrollo
* Evita tareas repetitivas y básicas
* Manejan la interfaz de usuario
* Brindan métodos y funciones generales
* Se aprende patrones de diseño y estilos de programación
* Los productos son mas eficientes

**1.4. Introducción A La Programación Orientada A Objetos (POO)**

La programación orientada a objetos es un paradigma o técnica de programación que fue inventada en <un poco de historia>

La Programación Orientada a Objetos (POO u OOP por sus siglas en ingles) es un paradigma de programación en el que los conceptos del mundo real relevantes para nuestro problema se modelan a través de clases y objetos, la gracia de la POO es que hace que los problemas sean mas sencillos, al permitir dividir el problema, esta división se la hace en objetos, de manera que cada objeto funcione de forma independiente, siendo este una parte del programa que posee sus datos y un funcionamiento propio, es decir un objeto esta formado por datos a los que llamaremos propiedades y métodos los que también son llamados procedimientos.

**Propiedad:** También conocidos como atributos, son variables que contienen datos asociados a un objeto, podemos decir que son las propiedades quienes guardan la información del objeto.

**Método:** Es un algoritmo asociado a un objeto, indica lo que el objeto esta en capacidad de hacer, podría concluir diciendo que el método son las funciones del objeto y que estas funciones trabajan con las propiedades, y que un objetyo puede tener tantos metodos como sea necsario.

**Ejemplo de objeto:**



Como podemos ver en la imagen, tenemos la grafica de un auto, de la cual vamos a definir sus propiedades y métodos.

Repasando los conceptos de propiedades podemos definir las siguientes:

1. Color
2. Tipo
3. Marca
4. Capacidad
5. Nro. Motor
6. Velocidades
7. Placa
8. Dueño
9. Combustible
10. Motor

Estos son apenas unas de las muchas propiedades que tiene este objeto, vamos a repasar cada una de ellas para verificar si son o no relevantes y además si cumplen su función que es la de dar información del objeto al que pertenecen.

**Color:** Se identifica el color del vehículo

**Tipo:** Seidentifica el tipo de auto que es

**Marca:** La marca por lo general es el nombre de la empresa que lo fabricó

**Capacidad:**Responde a la capacidad de carga o de pasajeros del vehículo vamos a asumir que es de 2 personas, aunque en la grafica vemos que con uno tiene.

**Nro. Motor:** Identificación única del motor, asignada por el fabricante.

**Velocidades:** La capacidad de la caja de cambios.

**Placa:** En nuestro pais para la libre circulación de los vehículos, estos deben contar con una identificación única llamada placa, la misma que es independiente del tipo de vehículo, marca, color, etc.

**Dueño:** A quien pertenece el vehículo, siempre va a tener un dueño.

**Combustible:** El tipo de combustible que usa ya sea gasolina. Diesel, Electricidad, etc.

**Transmisión:** En el mercado podemos encontrar principalmente dos tipos 4x4 y 4x2.

Vale la pena aclarar que más de un atributo por si solo no va a identificar al objeto, pero si pensamos en el conjunto de atributos lo más probable es que logremos identificar a un auto de un grupo de autos verdes de la misma marca.

**1.4.1. Elementos De La POO**

Hay más de un factor que determina la programación orientada a objetos que la identifican de otros paradigmas revisemos los principales:

**Abstracción**

Aísla a un elemento de su contexto, define las características esenciales de un objeto, con esto se quiere decir que es la virtualizacion de un objeto físico, luego de haberlo analizado, si pensamos en el auto la abstracción es el análisis que le hicimos.

**Encapsulamiento**

Reúne al mismo nivel de abstracción a todos los elementos que pueden considerase pertenecientes a una misma entidad, la encapsulación es la creación del objeto, el cual se obtuvo de la abstracción del objeto real con sus propiedades y métodos.

**Modularidad**

Característica que permite dividir a una aplicación en varios módulos o partes, las mismas que trabajan de forma independiente pero de manera colaborativa.

**Ocultación (Aislamiento)**

Las propiedades y métodos de los objetos están ocultas y son protegidas del exterior para que no sean modificadas por quienes no tengan acceso a ellas.

**Polimorfismo**

Es la capacidad que da a diferentes objetos del mismo tipo o instancias de la misma clase, la posibilidad de contar con métodos y propiedades con el mismo nombre sin que ello genere problemas entre un objeto y otro igual.

**Herencia**

Es una relación que existe entre dos o más clases, se denomina objeto padre al que posee los métodos y propiedades comunes y se les denomina hijos a los objetos que heredan dichos métodos y propiedades, la herencia ayuda mucho a la hora de implementar métodos generales para una serie de objetos, por ejemplo tenemos la clase vehículo que tendría métodos como frenar y acelerar y un objeto moto que herede de vehiculo esos metos y ademas tenga los suyos propios como cambio de velocidad.

**Recolección de Basura**

**encaspudamiento**

**1.4.2. Objetos**

**1.4.3. Clases**

**1.4.3.1. Instanciable (Clase Normal)**

**1.4.3.2. Heredada**

**1.4.3.3. Finales**

**1.4.3.4. Abstractas**

**1.4.3.5. Interfaces**

**1.4.4. Métodos**

**1.4.5. Evento Y Mensaje**

**1.4.6. Propiedades Y Atributos**

**1.5. Ejemplo Práctico**

**2. Capítulo II**

**2.1. INTRODUCCIÓN A HTML**

**2.2. Historia HTML**

**2.3. Versiones Y Tipos De HTML**

**2.4. Etiquetas**

**2.4.1. Definición Documento**

**2.4.2. Imágenes**

**2.4.3. Links**

**2.4.4. Capas**

**2.5. Formularios**

**2.6. Estructura**

**2.7. Ejemplo Pagina Básica HTML**

**2.8. ¿Qué es HTML5?**

**2.8.1. Nuevas Etiquetas**

**2.9. Introducción a CSS**

**2.9.1. Origen CSS**

**2.9.2. Usos CSS**

**2.10. Introducción Básica A CSS**

**2.10.1. Sintaxis Básica**

**2.10.2. Selectores**

**2.10.3. Propiedades**

**2.10.4. Primeros Pasos**

**2.10.5. Ejemplo De Una Pagina Con CSS**

**2.11. Introducción A Java Script**

**2.11.1. Origen Java Script**

**2.11.2. Usos Java Script**

**2.11.3. Lenguaje De Programación Java Script**

**2.11.4. Tipos De Datos**

**2.11.4.1. Numéricos**

**2.11.4.2. Cadenas**

**2.11.4.3. Booleanos**

**2.11.4.4. Nulos**

**2.11.4.5. Objeto**

**2.11.4.6. Arreglos**

**2.11.5. Sintaxis Básica Java Script**

**2.11.6. Operadores**

**2.11.7. Control De Flujo**

**2.11.7.1. Condicional If Else**

**2.11.7.2. Condicional Switch**

**2.11.7.3. Bucles For**

**2.11.7.4. Bucles While**

**2.11.7.5. Clases Y Funciones En Java Script**

**2.12. Importancia De Java Script En La Actualidad**

**2.13. Introducción A MySql**

**2.13.1. Comandos Básicos MySql**

**2.13.1.1.1. Comando SHOW**

**2.13.1.1.2. Crear Base De Datos Y Tablas**

**2.13.1.1.3. Alterar Bases De Datos**

**2.13.1.1.4. Consultas Con El Comando SELECT**

**2.13.1.1.4.1. Comprendiendo Al Comando SELECT**

**2.13.1.1.4.2. Consultas Básicas**

**2.13.1.1.5. Comando JOIN**

**2.13.1.1.5.1. Consultas Compuestas Con JOIN**

**2.13.1.1.6. Comando WHERE**

**2.13.1.1.6.1. Consultas Compuestas Con WHERE**

**2.13.1.2. Comandos Útiles MySql**

**2.13.1.3. Vistas**

**2.13.1.4. Procedimientos Almacenados**

**2.13.1.4.1. MySql Procedures**

**2.13.1.4.2. MySql Functions**

**2.13.1.4.3. MySql Triggers**

**2.14. Introducción a PHP**

**2.14.1. Referencia Del Lenguaje**

**2.14.2. Sintaxis Básica**

**2.14.3. Tipos**

**2.14.3.1. Booleanos**

**2.14.3.2. Enteros**

**2.14.3.3. Números De Punto Flotante**

**2.14.3.4. Cadenas**

**2.14.3.5. Arrays**

**2.14.3.6. Objetos**

**2.14.4. Variables**

**2.14.5. Constantes**

**2.14.6. Expresiones**

**2.14.7. Operadores**

**2.14.8. Estructuras de Control**

**2.14.9. Funciones**

**2.14.10. Clases Y Objetos**

**2.14.10.1. Introducción**

**2.14.10.2. Lo básico**

**2.14.10.3. Propiedades**

**2.14.10.4. Constantes De Clases**

**2.14.10.5. Autocarga De Clases**

**2.14.10.6. Constructores Y Destructores**

**2.14.10.7. Visibilidad**

**2.14.10.8. Herencia De Objetos**

**3. Capítulo III**

**3.1. Recolección De Información**

**3.1.1. Identificar Fuentes De Información**

**3.2. Definición De Problema. ¿Qué Es Un CMS?**

Es necesario conocer a perfección lo que es un CMS, con esto me refiero saber lo que es más allá de los conceptos empezaré citando la definición de Wikipedia la cual dice “*referencia wikipedia*”

Si vamos un poco mas alla también podemos concluir que un cms gestiona no solamente los contenidos que se muetran al los usuarios y administradores sino que además es encargado de gestionar los cmponentes internos de la aplicación, veamos los todos y cada uno de los ítems que manejan los CMS.

* Maneja las Imágenes

Se las puede manejar almacenándolas directamente en el servidor a través de en el que está alojado el sitio con la ayuda de un Uploader o caso contrario en servidores externos y rescatando el link del fichero.

Nota: Uploader es una aplicación que gestiona las imágenes, además en la segunda opción podemos perder la imagen con el tiempo porque la persona dueño del sitio tiene potestad sobre ellas y las puede eliminar si lo considera necesario.

* Manejar Videos

EL soporte de videos esta dado simplemente con mostrar videos de servidores externos el único a soportar será Youtube

* Google Custom Search (GCS)

Si pensamos un momento en el crecimiento del sitio, imaginemos por un momento que tenemos alojados más de cien artículos en más de diez categorías, sería muy difícil acceder a ellos solamente a través de los menús ya que estos serian muy extensos y además complicado para el usuario cuando este busque un contenido de su interés. Para enfrentar este inconveniente tenemos un producto de Google llamado Google Custom Search, lo que hace este sistema es identificar todo el contenido del sitio y lo indexa a Google, una vez implementado en la pagina podemos hacer búsquedas en el sitio con la ayuda de Google sin la necesidad de programar un buscador propio, y con toda la capacidad que Google posee.

El servicio que se va a elegir es el gratuito, este nos muestra publicidad en las búsquedas de los usuarios pero si se la desea eliminar de los resultados se debe pagar por el servicio, es bueno mencionar que esto ayuda mucho en el posicionamiento en el buscador de Google, aunque no es el único buscador que existe.

Nota: cuando el usuario se siente confundido o simplemente no encuentra lo que busca con facilidad abandona el sitio, hay que recordar que no somos los únicos que ofrecemos ese tipo de información, ha millones de sitios con igual o mejor información, por ene es importante priorizar el acceso a los contenidos.

* Google Analytics (GA)

Una vez que se pone en funcionamiento el sitio es necesario conocer sus estadísticas, la intención de este trabajo no es de mencionar cuantos servicios de estadísticas existen ni cual es el mejor, sino de recomendar uno que sea bueno y de preferencia gratuito, dentro de esta recomendación esta Google Analitics o GA, este no es solamente un contador de visitas, ayuda a la toma de decisiones para el rumbo del proyecto, ya que nos da información de que es lo que la gente mas lee en nuestro sitio, cuanto tiempo le dedica al sitio, desde donde esta viendo el sitio, que sistema operativo usa, su navegador, que tipo de dispositivo usa ya sea teléfonos tabletas u otros. Es un servicio muy completo, lo recomiendo.

* Google Maps

Es un servicio de mapas, lo vamos a usar en lo mas básico que es mostrar la ubicación de la empresa en la pagina de contactos.

* SEO Básico
* SEO es una técnica que hace que los contenidos de nuestro sitio web puedan ser encontrados por los buscadores. En cuanto SEO se van a seguir algunos parámetros específicos en la publicación de los artículos, como mencioné en una nota Google no es el único buscador que existe, pero es el mas grande, por eso controla la mayoría de los parámetros que los desarrolladores de contenido deben tener en cuenta para ser “vistos” por Google, y al ser tan grande esta empresa hizo que sus competidores como Yahoo Bing y muchos mas siguieran esos estándares para indexar los contenidos a sus bases de datos, esto se profundizará mas en la sección correspondiente.
* Redes Sociales

En la actualidad las redes sociales son una muy buena herramienta para atraer a personas, con esto nuestro sitio tiene publicidad a través de terceros de forma gratuita, permitiendo que los cibernautas conozcan nuestro sitio, hay un dicho que dice “*si quieres vender la gente debe saber que existes”*, esta es un muy buena forma de lograrlo, existes tres principales redes sociales como son.

Twitter

Facebook

Google Plus

**3.2.1. Definición De CMS**

**3.2.2. Historia De CMS**

**3.2.3. Tipos De CMS**

**3.2.3.1. Abiertos**

**3.2.3.2. Cerrados**

**3.2.3.3. Comerciales**

**3.3. Ventajas**

**3.4. Desventajas**

**3.5. Estudio Del Funcionamiento De Un CMS**

**3.5.1. Análisis De Las Prestaciones**

**3.5.2. Reconocimiento De La Estructura De Un CMS**

**3.5.3. Manejo De La Información Del Sitio Por Parte Del CMS**

**3.5.4. Almacenamiento De Datos**

**3.5.5. Manipulación De Datos Por El Usuario**

**3.5.6. Flujo De Datos**

**3.5.7. Conclusiones**

**4. Capítulo IV**

**4.1. Definición De Las Prestaciones Del Sistema**

**4.1.1. Análisis De Requerimientos**

**4.1.1.1. Hardware**

**4.1.1.2. Software**

**4.1.1.3. Personal Humano**

**4.1.2. Definición De Funcionalidades Del Sistema**

**4.2. Creación Del Modelo De Base De Datos**

**4.2.1. Creando El Primer Modelo**

**4.2.2. Sometiendo A Pruebas Modelo**

**4.2.3. Definición De Funciones Y Vistas**

**4.3. Análisis Diseño Lógico Del Sistema**

**4.3.1. Casos De Uso**

**4.3.2. Crear Diagramas De Modelo UML**

**4.4. Elaboración Completa Del Sistema En Diagramas**

**4.4.1. Diagrama De Bases De Datos**

**4.4.2. Diagrama De Clases**

**4.5. Dividir La Estructura Del Sistema**

**4.5.1. Introducción A Patrones De Diseño**

**4.5.2. Estructura Del Sistema Con Boilerplate**

**5. Capítulo V**

**5.1. Análisis General A Los Frameworks CodeIgniter Y Bootstrap**

**5.1.1. Propósitos**

**5.1.2. Estructura**

**5.1.3. Requerimientos**

**5.2. Introducción A CodeIgniter PHP**

**5.2.1. Diagrama de Flujo de la Aplicación**

**5.2.2. Modelo Vista Controlador**

**5.2.3. Instrucciones De Instalación**

**5.2.4. Configuración**

**5.2.5. Primeros Pasos**

**5.2.6. Temas Generales**

**5.2.6.1. URLS De CodeIgniter**

**5.2.6.2. Modelos**

**5.2.6.3. Vistas**

**5.2.6.4. Controladores**

**5.2.6.5. Usar Las Bibliotecas De CodeIgniter**

**5.2.6.6. Crear Bibliotecas En CodeIgniter**

**5.2.6.7. Ruteo URI**

**5.2.6.8. Funciones Comunes**

**5.2.6.9. Manejo de Errores**

**5.2.6.10. Seguridad**

**5.2.6.11. Estilo y Sintaxis Generales**

**5.2.7. Clases de CodeIgniter**

**5.2.7.1. Clase Config**

**5.2.7.2. Clase Database**

**5.2.7.3. Clase Email**

**5.2.7.4. Clase Encrypt**

**5.2.7.5. Clase Input**

**5.2.7.6. Clase Pagination**

**5.2.7.7. Clase Session**

**5.2.7.8. Clase Upload**

**5.2.8. Helpers de CodeIgniter**

**5.2.8.1. Helper Download**

**5.2.8.2. Helper Form**

**5.2.8.3. Helper URL**

**5.3. Introducción A Bootstrap CSS**

**5.3.1. Como Usar Bootstrap**

**5.3.2. Sintaxis De Bootstrap**

**5.3.3. Prestaciones Del Framework Bootstrap**

**5.3.4. Scaffolding**

**5.3.4.1. Grid System**

**5.3.4.2. Fluid Grid System**

**5.3.4.3. Layouts**

**5.3.5. Base CSS**

**5.3.5.1. Tables**

**5.3.5.2. Forms**

**5.3.5.3. Buttons**

**5.3.5.4. Images**

**5.3.5.5. Icons**

**5.3.6. Components**

**5.3.6.1. Dropdowns**

**5.3.6.2. Button Dropdowns**

**5.3.6.3. Nav**

**5.3.6.4. Navbar**

**5.3.6.5. Pagination**

**5.3.6.6. Labels**

**5.3.6.7. Alerts**

**5.3.7. Java Script**

**5.3.7.1. Modal**

**5.3.7.2. Tab**

**5.3.7.3. Tooltip**

**5.3.7.4. Collapse**

**6. Capítulo VI**

**6.1. Análisis De Interfaz De Usuario**

**6.1.1. Interfaz de Administrador**

**6.1.2. Interfaz de Usuario**

**6.2. Construcción De Interfaces**

**6.2.1. Construcción De Interfaz Administrador Con Bootstrap**

**6.2.2. Construcción De Interfaz Usuario Con Bootstrap**

**6.3. Pruebas De Interfaces**

**6.3.1. Implementación De Modelos**

**6.3.2. Implementación Modelo De Base Datos**

**6.3.3. Implementación Modelo De Clases**

**6.3.4. Implementación Diagrama General Del Sistema**

**6.4. Pruebas Del Sistema**

**6.5. Correcciones Al Sistema**

**6.6. Publicación Del Sistema**

**7. Capítulo VII**

**7.1. Conclusiones**

**7.2. Recomendaciones**

**7.3. Bibliografía**

**[1]** [**http://es.wikipedia.org/wiki/Internet**](http://es.wikipedia.org/wiki/Internet)

**[2]** **http://ellislab.com/codeigniter/user-guide/overview/appflow.html [3]**http://mautime.files.wordpress.com/2010/02/4378559350\_d12ac7a766\_b.jpg?w=720&h=347

**[4]** Van Gurp J. y Bosch J. 2001. Implementation and Evolution of Object-Oriented Frameworks: concepts & guidelines. Software: Practice andExperience. pp. 277-300

Pdf <http://observatoriodelacapacitacion.stps.gob.mx/oc/PDF/cursos_en_linea/POO_SE.pdf>