**Convenciones:**

* **Las referencias van en comillas y en cursiva**
* **El código fuente debe ir formateado y coloreado y con fuente monoespaciada**
* **Notas aparecen dentro de un recuadro gris**
* **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**

**1.1.1. Orígenes**

**1.1.2. Internet En La Actualidad**

**1.2. Definición De Sitios Web Y Su Estructura**

**1.2.1. Pagina Web Y Su Origen**

**1.2.2. Estructura**

**1.2.3. Importancia De Las Páginas Web**

**1.3. Introducción A los Frameworks**

**1.3.1. Introducción**

**1.3.2. Historia**

**1.3.3. Tipos De Frameworks**

**1.3.4. Importancia De Los Frameworks**

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

**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**

**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 Practico**

**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**