**PLAN DEL TRABAJO DE FIN DE CARRERA**

**Tema.**

Creación de un **CMS** (Content System Management) Gestor de contenidos con CodeIgniter y Bootstrap.

**Planteamiento del Problema:**

Desde hace mucho tiempo no basta con crear sitios web dinámicos, capaces de interactuar con el usuario, sino que además estos mismos, deben ser capaces de interactuar con el administrador, editores, participantes y demás roles. El problema a la hora de mantener el flujo de contenidos en un sitio web, está en que la persona que lo mantiene debe tener conocimientos de programación e informática, si pesamos un momento en la cantidad de sitios web que existen y la temática de cada uno de ellos, como sitios de ventas, de belleza, de cocina, hogar por solo mencionar a algunos, no todas las empresas o personas están en capacidad de mantener trabajando a un informático para sí, ya que es muy necesario a una persona con esos conocimientos.

Este no es un problema nuevo, tiene su origen con el aparecimiento del Internet el cual dio origen a los sitios web, visto la necesidad de mantener un sitio y economizar recursos, se crean los **CMS** (Content System Management) lo que se conoce como Gestor de Contenidos traducido al español, de lo que se encargan estas herramientas es de manejar los contenidos de manera transparente al usuario, de esta forma las entidades de una organización pueden administrar el sitio sin la necesidad de conocer nada de desarrollo web. Esto no es una herramienta nueva y existen muchos CMS en el mercado ya sean libres, gratuitos o de pago, cada uno con su filosofía y sus tecnologías, los gestores de contenidos están siempre alojados en el sitio web, y nos ayudan que sea soportado por todos los navegadores web, es el encargado de gestionar los contenidos y resolver problemas de compatibilidad con cada navegador (esto solo si el diseñador del sitio le dio esa capacidad)

**Objetivos Del Trabajo.**

Objetivo General:

* Implementar un CMS (Content System Management) práctico, sencillo y fácil de usar con herramientas libres, capaz de manejar el sitio de una persona o empresa sin la necesidad de conocimientos avanzados de informática.

Objetivos Específicos:

* Desarrollar y Definir el Funcionamiento de un CMS
* Implementar el uso de Frameworks, para un desarrollo rápido y robusto
* Implementar modelos de desarrollo como MVC (modelo vista controlador)
* Crear una interfaz independiente del contenido y viceversa
* Conocer Tecnologías nuevas como HTML5 CSS3 y Java Script

**Justificación del Trabajo:**

Para vivir en Internet o incluso de él, y avanzar conforme a los requerimientos no solo de los clientes sino de las autoridades, es preciso responder con prontitud y certeza a sus demandas, con esto quiero decir:

Que aunque se tuviese a un desarrollador web propio, este no está en la capacidad de responder inmediatamente todos los requerimientos, por ejemplo. Si en un sitio web dedicado al comercio electrónico formado por varias personas, si alguien desease cambiar la descripción de un producto o agregar imágenes, no lo puede hacer directamente, tendría que buscar al profesional encargado, y aún para el desarrollador por mas experimentado y hábil que sea, va a tomar tiempo en resolverlo, además no siempre va a estar disponible y aunque exista más de un desarrollador trabajando en turnos rotativos, estos solo pueden atender una petición a la vez eso significa que están en capacidad de atender a una persona por turno, imaginemos el tiempo que tomaría editar y publicar 300 productos, además hay que considerar los costos ya que un solo desarrollador no es suficiente, entonces fuese más fácil si el mismo usuario puede modificar las características de su publicación cuando lo quiera desde donde sea y a la hora que fuese, ya que es sistema estaría trabajando 24/7 simultáneamente para todos, sin la necesidad del desarrollador, sin embargo no pretendo desvalorizar al desarrollador o empresas dedicadas a este negocio ya que es indispensable que alguna entidad le brinde soporte, ya que nada es infalible, sobre todo las cosas hechas por el hombre.

Concluyo recalcando que un sistema CMS(Content System Management) es una muy buena solución para que personas que no conocen de desarrollo web puedan publicar contenidos de cualquier índole en internet, creando de esta forma un internet más accesible para todos, y que las personas que visitan el sitio puedan recibir esa información conforme los deseos y gustos de autor.

Justificación Metodológica.

El fin de este trabajo es crear un producto óptimo, que responda a las necesidades de la entidad que lo vaya a usar, para descubrir las necesidades de esa entidad nos sujetaremos a la definición de un CMS (Content System Management) y a las prestaciones que este debe tener, y de los diseños de interfaz a los que el administrador, usuario y demás se van a enfrentar durante el uso del sistema, por esa razón deben ser amigables y fáciles de usar, los pasos a seguir son los siguientes:

* Investigar el funcionamiento de un CMS a detalle
* Establecer un modelo de bases de datos que responda al análisis anterior
* Comprobar modelo de base de datos y realizar las correcciones pertinentes
* Grafica borrador del funcionamiento del sistema, implementando todas las prestaciones que este contenga.
* Análisis por separado de las funcionalidades del sistema, junto con documentación y diagramas, como resultado tendremos el problema dividido en entidades independientes que trabajan entre sí.
* Grafica Final del funcionamiento del sistema
* Elaboración de Estructura informática del sistema, como lo son patrones de diseño, arquitectura informática, convenciones de programación.
* Comprobar que la estructura sea eficiente y robusta, con esto lo que está buscando que la aplicación funcione en todos los navegadores, Sistema Operativo o plataforma.
* Implementar las funcionalidades de sistema dividiéndolas en módulos o clases en programación orientada a objetos, esto se llevará a cabo con la ayuda de CodeIgniter.
* Se trabaja en el diseño de la interfaz de administrador y usuario implementado a Bootstrap Frameworks CSS
* Se combina el diseño grafico con el diseño lógico del sistema y se implementa la aplicación.
* Se realizan pruebas de funcionamiento a todas las prestaciones de la aplicación al igual que a la interfaz de usuario ya que esta debe ser lo más sencilla posible.
* Se sube la aplicación a un servidor web, en mi caso estará alojado en http://isp.liposerve.com

**Temario.**

**Capítulo I**

1. Definición Y Origen De Internet
   1. Definición
   2. Orígenes
   3. Internet En La Actualidad
2. Definición De Sitios Web Y Su Estructura
   1. Pagina Web Y Su Origen
   2. Estructura
   3. Importancia De Las Páginas Web
3. Introducción A Frameworks
   1. Introducción
   2. Historia
   3. Tipos De Frameworks
   4. Importancia De Los Frameworks
   5. ¿Por qué usar un framework?
4. Introducción A La Programación Orientada A Objetos (Poo)
   1. Elementos De La Poo
      1. Objetos
      2. Clases
         1. Instanciable (Clase Normal)
         2. Heredada
         3. Finales
         4. Abstractas
         5. Interfaces
      3. Métodos
      4. Evento Y Mensaje
      5. Propiedades Y Atributos
   2. Ejemplo Practico

**Capítulo II**

1. INTRODUCCIÓN A HTML
   1. Historia HTML
   2. Versiones Y Tipos De HTML
   3. Etiquetas
      1. Definición Documento
      2. Imágenes
      3. Links
      4. Capas
      5. Formularios
   4. Estructura
   5. Ejemplo Pagina Básica HTML
2. ¿Qué es HTML5?
   1. Nuevas Etiquetas
3. Introducción a CSS
   1. Origen CSS
   2. Usos CSS
   3. Introducción Básica A CSS
      1. Sintaxis Básica
      2. Selectores
      3. Propiedades
      4. Primeros Pasos
      5. Ejemplo De Una Pagina Con CSS
4. Introducción A Java Script
   1. Origen Java Script
   2. Usos Java Script
   3. Lenguaje De Programación Java Script
      1. Tipos De Datos
         1. Numéricos
         2. Cadenas
         3. Booleanos
         4. Nulos
         5. Objeto
         6. Arreglos
      2. Sintaxis Básica Java Script
      3. Operadores
      4. Control De Flujo
         1. Condicional If Else
         2. Condicional Switch
         3. Bucles For
         4. Bucles While
      5. Clases Y Funciones En Java Script
      6. Importancia De Java Script En La Actualidad
5. Introducción A MySql
   1. Comandos Básicos MySql
      1. Comando SHOW
      2. Crear Base De Datos Y Tablas
      3. Alterar Bases De Datos
      4. Consultas Con El Comando SELECT
         1. Comprendiendo Al Comando SELECT
         2. Consultas Básicas
      5. Comando JOIN
         1. Consultas Compuestas Con JOIN
      6. Comando WHERE
         1. Consultas Compuestas Con WHERE
      7. Comandos Útiles MySql
      8. Vistas
      9. Procedimientos Almacenados
         1. MySql Procedures
         2. MySql Functions
         3. MySql Triggers
6. Introducción a PHP
   1. Referencia Del Lenguaje
      1. Sintaxis Básica
      2. Tipos
         1. Booleanos
         2. Enteros
         3. Números De Punto Flotante
         4. Cadenas
         5. Arrays
         6. Objetos
      3. Variables
      4. Constantes
      5. Expresiones
      6. Operadores
      7. Estructuras de Control
      8. Funciones
      9. Clases Y Objetos
         1. Introducción
         2. Lo básico
         3. Propiedades
         4. Constantes De Clases
         5. Autocarga De Clases
         6. Constructores Y Destructores
         7. Visibilidad
         8. Herencia De Objetos

**Capítulo III**

1. Recolección De Información
   1. Identificar Fuentes De Información
2. Definición De Problema. ¿Qué Es Un CMS?
   1. Definición De CMS
   2. Historia De CMS
   3. Tipos De CMS
      1. Abiertos
      2. Cerrados
      3. Comerciales
   4. Ventajas
   5. Desventajas
3. Estudio Del Funcionamiento De Un CMS
   1. Análisis De Las Prestaciones
   2. Reconocimiento De La Estructura De Un CMS
   3. Manejo De La Información Del Sitio Por Parte Del CMS
   4. Almacenamiento De Datos
   5. Manipulación De Datos Por El Usuario
   6. Flujo De Datos
   7. Conclusiones

**Capítulo IV**

1. Definición De Las Prestaciones Del Sistema
   1. Análisis De Requerimientos
      1. Hardware
      2. Software
      3. Personal Humano
   2. Definición De Funcionalidades Del Sistema
2. Creación Del Modelo De Base De Datos
   1. Creando El Primer Modelo
   2. Sometiendo A Pruebas Modelo
   3. Definición De Funciones Y Vistas
3. Análisis Diseño Lógico Del Sistema
   1. Casos De Uso
   2. Crear Diagramas De Modelo UML
   3. Elaboración Completa Del Sistema En Diagramas
      1. Diagrama De Bases De Datos
      2. Diagrama De Clases
4. Dividir La Estructura Del Sistema
   1. Introducción A Patrones De Diseño
   2. Estructura Del Sistema Con Boilerplate

**Capítulo V**

1. Análisis General A Los Frameworks CodeIgniter Y Bootstrap
   1. Propósitos
   2. Estructura
   3. Requerimientos
2. Introducción A CodeIgniter PHP
   1. Diagrama de Flujo de la Aplicación
   2. Modelo Vista Controlador
   3. Instrucciones De Instalación
   4. Configuración
   5. Primeros Pasos
   6. Temas Generales
      1. URLS De CodeIgniter
      2. Modelos
      3. Vistas
      4. Controladores
      5. Usar Las Bibliotecas De CodeIgniter
      6. Crear Bibliotecas En CodeIgniter
      7. Ruteo URI
      8. Funciones Comunes
      9. Manejo de Errores
      10. Seguridad
      11. Estilo y Sintaxis Generales
   7. Clases de CodeIgniter
      1. Clase Config
      2. Clase Database
      3. Clase Email
      4. Clase Encrypt
      5. Clase Input
      6. Clase Pagination
      7. Clase Session
      8. Clase Upload
   8. Helpers de CodeIgniter
      1. Helper Download
      2. Helper Form
      3. Helper URL
3. Introducción A Bootstrap CSS
   1. Como Usar Bootstrap
   2. Sintaxis De Bootstrap
   3. Prestaciones Del Framework Bootstrap
      1. Scaffolding
         1. Grid System
         2. Fluid Grid System
         3. Layouts
      2. Base CSS
         1. Tables
         2. Forms
         3. Buttons
         4. Images
         5. Icons
      3. Components
         1. Dropdowns
         2. Button Dropdowns
         3. Nav
         4. Navbar
         5. Pagination
         6. Labels
         7. Alerts
      4. Java Script
         1. Modal
         2. Tab
         3. Tooltip
         4. Collapse

**Capítulo VI**

1. Análisis De Interfaz De Usuario
   1. Interfaz de Administrador
   2. Interfaz de Usuario
2. Construcción De Interfaces
   1. Construcción De Interfaz Administrador Con Bootstrap
   2. Construcción De Interfaz Usuario Con Bootstrap
3. Pruebas De Interfaces

**Capítulo VII**

1. Implementación De Modelos
   1. Implementación Modelo De Base Datos
   2. Implementación Modelo De Clases
   3. Implementación Diagrama General Del Sistema
2. Pruebas Del Sistema
3. Correcciones Al Sistema
4. Publicación Del Sistema

**Capítulo VIII**

1. Conclusiones
2. Recomendaciones
3. Bibliografía

**Presupuesto y Fuentes de Financiamiento.**

Presupuesto:

Este es un proyecto que no está dirigido a una persona o empresa sino a un grupo de entidades, por lo cual el presupuesto no existe para la elaboración del producto ya que la mayoría de herramientas y tecnologías usadas para crear el producto son libres y gratuitas, lo que si sería de considerar es el costo del trabajo hecho por el desarrollador, pero dada la naturaleza del trabajo este es un costo que podemos omitir. El presupuesto es más bien para el funcionamiento del sistema, el cual tiene unos costos que hay que cubrir los mismos que detallo a continuación:

|  |  |
| --- | --- |
| Descripción | Costo USD |
| Registro del Dominio 1 año | 11.00 |
| Hosting 1 año | 29.00 |
| TOTAL | 39.00 |

Podemos concluir diciendo que un año de funcionamiento nos cuesta **43.68** **USD** con impuestos incluidos (Datos consultados en Ecuahosting.com), estos costos pueden ser más elevados ya que en este caso no se piensa recibir un nivel alto de tráfico, en caso que el proyecto prospere ese valor se incrementaría a mas o menos 50 USD por mes.

Financiamiento:

Si se desea poner el proyecto en producción, es necesario buscar financiamiento, pero en mi caso eso no es así, ya que dispongo de servidores en internet, uno de los cuales voy a usar para poner mi proyecto en marcha, la dirección web del proyecto será http://isp.liposerve.com.

**Cronograma de Actividades.**

* Recolectar información referente a mi tema
* Análisis de la información
* Generar modelos a partir de esa información
* Someter a prueba los modelos
* Realizar correcciones a los modelos
* Elaborar diagramas de Casos de Uso
* Elaborar diagramas de clases
* Elaborar diagrama general del sistema
* Imprentar diagrama de base de datos
* Ingresar datos a la base de datos
* Realizar correcciones en caso de encontrar fallas en el momento de ingresar la información
* Implementar funciones vistas y disparadores a la base de datos
* Generar la estructura básica del sistema con la ayuda de Boilerplate y CodeIgniter
* Trabajar en el requerimiento de las vistas
* Maquetar las vistas de inicio, de presentación de contenidos, pagina 404 y la de contacto, teniendo presente la presentación de errores y advertencias que se hacen al usuario y administrador
* Análisis de códigos similares en las vistas, el cual es aislado en archivos diferentes para luego proceder a ármalos como si se tratase de un rompecabezas.
* Generar el Sistema y someter a pruebas, realizado al mismo tiempo la corrección de las mismas
* Publicación del Sistema