Universidad Autónoma del Estado de Hidalgo

Instituto de Ciencias Básicas e Ingeniería

Área Académica de Computación

Licenciatura en Sistemas Computacionales

**Aplicación online para creación y administración de sitios web corporativos orientados a PyMES.**

**Caso de Estudio: tentupagina.com**

**P R E S E N T A**

VARGAS MORENO ADALBERTO.

**Asesor:**

L.S.C. Arturo Ocampo López.

**Catedrática:**

M.C.C. Verónica Martínez Lazcano

Mineral de la Reforma, Hidalgo, Mayo de 2014

México

Resumen

En el presente documento se realiza el análisis del desarrollo de un sistema de información, para la generación automatizada de sitios web orientados a PyMES, así mismo se hace énfasis en el uso de recursos tecnológicos emergentes para agilizar el desarrollo de la mano de la metodología empleada en el proyecto.

Contenido

[Antecedentes y definición del problema I](#_Toc389021546)

[Propuesta de solución II](#_Toc389021547)

[Objetivo general II](#_Toc389021548)

[Objetivos específicos II](#_Toc389021549)

[Justificación III](#_Toc389021550)

[Introducción III](#_Toc389021551)

[I. CAPÍTULO I: MARCO TEÓRICO 1](#_Toc389021552)

[1.1. Introducción a los lenguajes de programación 1](#_Toc389021553)

[1.1.1. Lenguajes de alto nivel 1](#_Toc389021554)

[1.1.2 Clasificación de los lenguajes de programación 2](#_Toc389021555)

[1.2. Lenguajes por lugar de ejecución 7](#_Toc389021556)

[1.2.1. Lenguaje PHP 7](#_Toc389021557)

[1.2.2. JavaScript 8](#_Toc389021558)

[1.2.3. Librería JQuery 8](#_Toc389021559)

[1.2.4. JavaScript Object Notation 9](#_Toc389021560)

[1.2.5. Hojas de estilo de Cascada 9](#_Toc389021561)

[1.2.6. Lenguaje HTML 9](#_Toc389021562)

[1.3. Arquitectura 9](#_Toc389021563)

[1.3.1. Estructura cliente servidor 10](#_Toc389021564)

[1.3.2. Modelo Vista Controlador 10](#_Toc389021565)

[1.4. Frameworks para desarrollo ágil 10](#_Toc389021566)

[1.4.1. Framework Back-end 11](#_Toc389021567)

[1.4.2. Framework Front-End 11](#_Toc389021568)

[1.5. Framework Laravel 11](#_Toc389021569)

[1.6. Framework Bootstrap 12](#_Toc389021570)

[1.6.1. Características 12](#_Toc389021571)

[1.7. Introducción a las bases de datos 12](#_Toc389021572)

[1.7.1. Base de datos relacionales 12](#_Toc389021573)

[1.7.2. MySQL 13](#_Toc389021574)

[1.8. Sistema de información 13](#_Toc389021575)

[II. CAPÍTULO II: METODOLOGÍA 14](#_Toc389021576)

[2.1. Introducción a la metodología 14](#_Toc389021577)

[2.1.1. Características 14](#_Toc389021578)

[2.2. Aplicación de la metodología 14](#_Toc389021579)

[2.2.1. Comunicación 15](#_Toc389021580)

[2.2.2. Simplicidad 15](#_Toc389021581)

[2.2.3. Retroalimentación 15](#_Toc389021582)

[2.2.4. Respeto 15](#_Toc389021583)

[2.2.5. Coraje 15](#_Toc389021584)

[III. CAPÍTULO III: ESTADO DEL ARTE 16](#_Toc389021585)

[3.1. Introducción a proyectos alternos 16](#_Toc389021586)

[ESTADO DEL ARTE 17](#_Toc389021587)

[IV. CAPÍTULO IV: HERRAMIENTAS COMPUTACIONALES 19](#_Toc389021588)

[Herramientas a utilizar en el proyecto 19](#_Toc389021589)

[V. CAPÍTULO V: ANÁLISIS Y DISEÑO DEL SISTEMA 21](#_Toc389021590)

[Metas 21](#_Toc389021591)

[Hipótesis 21](#_Toc389021592)

[Alcances y limitaciones 22](#_Toc389021593)

[Propuesta de solución 23](#_Toc389021594)

[Mapas de navegación 24](#_Toc389021595)

[Casos de uso 25](#_Toc389021596)

[Prototipo 26](#_Toc389021597)

[VI. CAPÍTULO VI: DESARROLLO DEL SISTEMA 28](#_Toc389021598)

[Desarrollo 28](#_Toc389021599)

[Creación de formulario de registro 30](#_Toc389021600)

[Comienzo de desarrollo de panel 30](#_Toc389021601)

[Definición de estilos CSS (20 líneas de 7098) 31](#_Toc389021602)

[Responsive CSS (20 de 415 líneas de código) 32](#_Toc389021603)

[Integración de repositorio de librerias JavaScript 33](#_Toc389021604)

[Generación de vista del panel (20 de 146 líneas) 34](#_Toc389021605)

[Maquetado del front de panel 35](#_Toc389021606)

[Edición de Json RESTful de 20 de 177 líneas 36](#_Toc389021607)

[Conexión de Json con admin (18 de 95 líneas) 37](#_Toc389021608)

[VII. CAPÍTULO VII: PRUEBAS DEL SISTEMA 38](#_Toc389021609)

[Conclusiones 39](#_Toc389021610)

[Anexo I: Cronograma de actividades 40](#_Toc389021611)

[Anexo II: Diapositivas 43](#_Toc389021612)

[Glosario 44](#_Toc389021613)

[Referencias 45](#_Toc389021614)

Ilustraciones

[Ilustración V.1: Estructura de los sitios web autogenerados 24](#_Toc389021615)

[Ilustración V.2: Estructura del panel de administración del sitio web 24](#_Toc389021616)

[Ilustración V.3: Estructura de la página de registro para acceder al servicio 25](#_Toc389021617)

[Ilustración V.4: Diagrama de casos de uso de la aplicación web de tentupagina.com 26](#_Toc389021618)

[Ilustración V.5: *Boceto Panel de administración sitio web. Pantalla con panel de control de página web.* 27](#_Toc389021619)

[Ilustración V.6: *Boceto de sitio web. Aspecto de sitio web generado mediante la aplicación, como se puede observar es un sitio web de tipo corporativo informativo.* 27](#_Toc389021620)

[Ilustración VI.1: Github logo, el usar control de versiones es una buena práctica en desarrollo de software 29](#_Toc389021621)

# Antecedentes y definición del problema

Tentupagina.com es un sitio web informativo con el propósito de presentar al público una gama de servicios para diseño y desarrollo web. Dentro dicho portal web sólo es posible consultar información. La mayoría de las solicitudes son para crear sitios web corporativos para PyMES. Los sitios web son adquiridos por medio de pedidos mediante un formato que pide los datos principales de contacto. Una solicitud de servicio para desarrollo de un sitio conlleva a planificar una cita, hacer la visita al prospecto, hacer la oferta, colectar información, hacer una propuesta, cerrar la venta, definir fechas de entrega y dar el mantenimiento al sitio.

El problema principal es que el modo de operación, requiere de la intervención de personal humano para atender cada cita solicitada, es necesario dedicar tiempo para cada proyecto, las entregas no son inmediatas y además el costo del servicio de diseño web no puede ser reducido. Otro problema es el tiempo necesario a invertir en diseño y desarrollo de cada proyecto, debido a que es requerido hacer entregas prototipo y final, esto se traduce en una mayor inversión de esfuerzo por cada sitio web nuevo. Actualmente para el desarrollo de un sitio web por cada visita al cliente ya sea para ofertar servicios, concretar una venta, colectar información contenido del sitio o para hacer entregas; se necesita cubrir gastos de transporte, gastos por concepto de llamadas telefónicas y tener disponibilidad de tiempo.

El usuario entra al portal informativo y captura sus datos de contacto en un formato en el que ingresa su nombre, correo electrónico, teléfono y además se cuenta con un campo para la redacción de información adicional. Posteriormente dicho mensaje es enviado a una bandeja de entrada de correo electrónico de tentupagina.com, éste mansaje es consultado, y con los datos de contacto recabados se hace contacto entre tentupagina.com y el prospecto cliente, con lo que se concreta una cita para presencialmente intentar vender los servicios, si el prospecto continúa interesado en los servicios de tentupagina.com, se agenda la fecha de una segunda cita para recabar la información a incorporar en el sitio web. A partir de la fecha de colecta de información es cuando se procede a hacer el diseño y codificación del mismo en lenguajes web, esto para posteriormente, hacer la entrega del prototipo de sitio web. Una vez que es aprobado dicho prototipo, con base en éste se continúa el desarrollo del proyecto hasta concluirlo. Una vez terminado el proyecto, se importa en un servidor web para que el nuevo sitio web pueda ser consultado desde una URL pública, se hace la entrega del proyecto al cliente y se le da mantenimiento al sitio para que permanezca operativo.

# Propuesta de solución

Después de conocer los antecedentes y la problemática, se propone el desarrollo de un sistema en línea de apoyo a la creación y administración de sitios web corporativos orientados a PyMES, que permita optimizar el manejo y el control de cada nuevo sitio web, facilitando a cada usuario crear y gestionar el sitio web de una o varias PyME en tiempo real.

# Objetivo general

Extender el portal web de tentupagina.com mediante la adición de una aplicación online que permita a sus usuarios micros empresarios crear, modificar y publicar sitios web, haciendo uso de cualquier dispositivo que incorpore conexión a internet y tenga navegador web; con el objeto de facilitar el mantenimiento, actualización y control sobre cada sitio web generado mediante el uso de dicho sistema informático.

# Objetivos específicos

* Diseñar la arquitectura de un sistema que permita optimizar los procesos de colecta de información, creación, diseño y codificación de sitios web corporativos
* Crear una base de datos para almacenar información sensible y para el funcionamiento del sistema online.
* Crear el repositorio de datos para toda la información colectada de cada sitio web, esto para no saturar de consultas a la base datos.
* Diseñar interfaces del sistema simples, de modo que para el usuario sea fácil entender y utilizar el sistema de información.
* Codificar el sistema con base en el diseño previamente realizado.
* Instalar el sistema en el servidor web.

# Justificación

Con el desarrollo del sistema online para creación y administración de sitios web corporativos orientados a PyMES, se pretende que desde el portal web tentupagina.com, se puedan mejorar los procesos de colecta de información, diseño y codificación de cada sitio web solicitado, esto con el propósito de brindar un servicio eficiente a cada usuario, además de mejorar notablemente los tiempos de respuesta entre los procesos necesarios.

# Introducción

El presente documento muestra la información que da seguimiento al desarrollo de un sistema informático con el propósito de optimizar los procesos de colecta de información, codificación y diseño web.

Dentro del capítulo I, se intenta exponer una breve explicación de lo que es la programación y los fundamentos teóricos necesarios para el desarrollo de un sistema operable en ambiente web, por lo que se define lenguaje de programación, continuando con programación orientada a objetos y lenguajes de etiquetado.

A continuación se explica el patrón de arquitectura de Software MVC y los principales Frameworks para desarrollo ágil de aplicaciones, para continuar con una reseña de los Frameworks Laravel y Bootstrap, además de conceptos y terminología para desarrollar usando tales herramientas computacionales. Posteriormente se explica lo que es una base de datos, los modelos existentes y la creación y diseño de éstas.

En el capítulo II, se mencionan textualmente los principales pasos del proceso del desarrollo de Sistemas de Información. Se exhibe el análisis del modelo de negocio de tentupagina.com. Se expone la problemática planteada por parte de un grupo de usuarios de los servicios exhibidos en el portal web de tentupagina.com

Además se describe el proceso de construcción de micro portales web, el proceso actual para recopilar la información a incorporar en los nuevos sitios web y el flujo de los datos proporcionados hasta ser maquetados. Con base en el análisis de lo anterior, se plantea el desarrollo de un sistema que satisfaga las necesidades expuestas, se incluye el diseño lógico del mismo y la base de datos a utilizar, así como también el repositorio de datos usado como extensión de ésta. Se muestra el diseño de los diagramas de cada módulo que constituye a la aplicación web.

En el capítulo III, se describe la fase de desarrollo, diseño y de codificación de los componentes del sistema, se incluye además la descripción de los métodos y propiedades de algunos de los controladores más importantes que en conjunto, integran el sistema informático.

Supremamente en el capítulo IV se describen las primeras pruebas realizadas al sistema, además se resumen los resultados obtenidos en las mismas, así como algunas observaciones y conclusiones respecto a su funcionamiento; en base a los resultados obtenidos, se determinará si el sistema cumple con los objetivos planteados y los requerimientos de los usuarios del sistema de información.

# CAPÍTULO I: MARCO TEÓRICO

Este capítulo intenta introducir al lector al marco de programación y desarrollo de software, para ello se inicia con la definición de lenguaje de programación, continuando con una breve explicación sobre los lenguajes de alto nivel.

Consecutivamente se indica de manera sencilla en qué consisten los lenguajes orientados a objetos, para continuar con una revisión de PHP, se explicará el patrón de arquitectura de software MVC, para posteriormente dar una revisión a Framework Laravel, además de conceptos y terminología necesarios para programar usando éste Framework.

Finalmente se explica lo que es una base de datos, repositorio de datos, principales conceptos y terminología, para comprender la importancia del uso de estas como columna vertebral en los sistemas de información.

## Introducción a los lenguajes de programación

El hombre, llevado por la necesidad de facilitar le ejecución de tareas tediosas y cansadas de las personas, ha inventado multitud de herramientas a lo largo de la historia, con el afán de mejorar la calidad de vida. Entre estas invenciones tenemos las computadoras, que tienen la peculiaridad de poder ser programables, es decir; recibir instrucciones para que posteriormente de manera autónoma (sin la intervención de ente alguno) ejecute las tareas que le fueron asignadas, siguiendo un procedimiento previamente definido. Debido a la necesidad de dar instrucciones a las computadoras, para posteriormente automatizar procesos; surgen los lenguajes de programación.

Los lenguajes de programación son un vocabulario y un conjunto de reglas gramaticales para instruir a una computadora para realizar tareas específicas. Tenemos por ejemplos de lenguajes a BASIC, C, C + +, COBOL, FORTRAN, Ada, Pascal, Java y PHP. Cada lenguaje tiene un conjunto único de palabras clave y una sintaxis especial para la organización de las instrucciones del programa. Los lenguajes de programación pueden dividirse en dos grandes grupos; lenguajes de programación de alto y bajo nivel

### Lenguajes de alto nivel

Un lenguaje de programación de alto nivel es un lenguaje de programación con una fuerte abstracción de los detalles de la computadora. En comparación con los lenguajes de programación de bajo nivel, se pueden utilizar los elementos de lenguaje natural, será más fácil de usar, o puede automatizar áreas importantes de los sistemas de computación (por ejemplo, gestión de memoria), haciendo que el desarrollo de un programa sea más simple y más comprensible en relación con un lenguaje de bajo nivel.

### 1.1.2 Clasificación de los lenguajes de programación

La existencia de una amplia gama de lenguajes de programación, propicia que cada uno de ellos esté encaminado a resolver ciertas tareas, dentro de la amplia problemática de la explotación de información, o bien, a que su arquitectura, o su forma de llevar a cabo la programación tiene un enfoque particular. A continuación se describe la clasificación de los lenguajes de programación con base en 12 criterios según el portal web Qbitacora (Qbitacora, 2014).

**1. Nivel de abstracción.**

Según el nivel de abstracción, o sea, según el grado de cercanía a la máquina:

* Lenguajes de bajo nivel: La programación se realiza teniendo muy en cuenta las características del procesador. Ejemplo: Lenguajes ensamblador.
* Lenguajes de nivel medio: Permiten un mayor grado de abstracción pero al mismo tiempo mantienen algunas cualidades de los lenguajes de bajo nivel. Ejemplo: C puede realizar operaciones lógicas y de desplazamiento con bits, tratar todos los tipos de datos como lo que son en realidad a bajo nivel (números), etc.
* Lenguajes de alto nivel: Más parecidos al lenguaje humano. Manejan conceptos, tipos de datos, etc., de una manera cercana al pensamiento humano ignorando (abstrayéndose) del funcionamiento de la máquina. Ejemplos: Java, Ruby.

Hay quien sólo considera lenguajes de bajo nivel y de alto nivel, (en ese caso, C es considerado de alto nivel).

**2. Propósito.**

Según el propósito, es decir, el tipo de problemas a tratar con ellos:

* Lenguajes de propósito general: Aptos para todo tipo de tareas: Ejemplo: C.
* Lenguajes de propósito específico: Hechos para un objetivo muy concreto. Ejemplo: Csound (para crear ficheros de audio).
* Lenguajes de programación de sistemas: Diseñados para realizar sistemas operativos o drivers. Ejemplo: C.
* Lenguajes de script: Para realizar tareas varias de control y auxiliares. Antiguamente eran los llamados lenguajes de procesamiento por lotes (batch) o JCL (“Job Control Languages”). Se subdividen en varias clases (de shell, de GUI, de programación web, etc.). Ejemplos: bash (shell), mIRC script, JavaScript (programación web).

**3. Evolución histórica.**

Con el paso del tiempo, se va incrementando el nivel de abstracción, pero en la práctica, los de una generación no terminan de sustituir a los de la anterior:

* Lenguajes de primera generación (1GL): Código máquina.
* Lenguajes de segunda generación (2GL): Lenguajes ensamblador.
* Lenguajes de tercera generación (3GL): La mayoría de los lenguajes modernos, diseñados para facilitar la programación a los humanos. Ejemplos: C, Java.
* Lenguajes de cuarta generación (4GL): Diseñados con un propósito concreto, o sea, para abordar un tipo concreto de problemas. Ejemplos: NATURAL, Mathematica.
* Lenguajes de quinta generación (5GL): La intención es que el programador establezca el qué problema ha de ser resuelto y las condiciones a reunir, y la máquina lo resuelve. Se usan en inteligencia artificial. Ejemplo: Prolog.

**4. Manera de ejecutarse.**

* Lenguajes compilados: Un programa traductor traduce el código del programa (código fuente) en código máquina (código objeto). Otro programa, el enlazador, unirá los ficheros de código objeto del programa principal con los de las librerías para producir el programa ejecutable. Ejemplo: C.
* Lenguajes interpretados: Un programa (intérprete), ejecuta las instrucciones del programa de manera directa. Ejemplo: Lisp.
* Mixtos, como Java, que primero pasan por una fase de compilación en la que el código fuente se transforma en “bytecode”, y este “bytecode” puede ser ejecutado luego (interpretado) en ordenadores con distintas arquitecturas (procesadores) que tengan todos instalados la misma “máquina virtual” Java.

**5. Manera de abordar la tarea a realizar.**

Según la manera de abordar la tarea a realizar, pueden ser:

* Lenguajes imperativos: Indican cómo hay que hacer la tarea, es decir, expresan los pasos a realizar. Ejemplo: C.
* Lenguajes declarativos: Indican qué hay que hacer. Ejemplos: Lisp, Prolog. Otros ejemplos de lenguajes declarativos, pero que no son lenguajes de programación, son HTML (para describir páginas web) o SQL (para consultar bases de datos).

**6. Paradigma de programación.**

El paradigma de programación es el estilo de programación empleado. Algunos lenguajes soportan varios paradigmas, y otros sólo uno. Se puede decir que históricamente han ido apareciendo para facilitar la tarea de programar según el tipo de problema a abordar, o para facilitar el mantenimiento del software, o por otra cuestión similar, por lo que todos corresponden a lenguajes de alto nivel (o nivel medio), estando los lenguajes ensambladores “atados” a la arquitectura de su procesador correspondiente. Los principales son:

* Lenguajes de programación procedural: Divide el problema en partes más pequeñas, que serán realizadas por subprogramas (subrutinas, funciones, procedimientos), que se llaman unas a otras para ser ejecutadas. Ejemplos: C, Pascal.
* Lenguajes de programación orientada a objetos: Crean un sistema de clases y objetos siguiendo el ejemplo del mundo real, en el que unos objetos realizan acciones y se comunican con otros objetos. Ejemplos: C++, Java.
* Lenguajes de programación funcional: La tarea se realiza evaluando funciones, (como en Matemáticas), de manera recursiva. Ejemplo: Lisp.
* Lenguajes de programación lógica: La tarea a realizar se expresa empleando lógica formal matemática. Expresa qué computar. Ejemplo: Prolog.

**7. Lugar de ejecución.**

En sistemas distribuidos, según dónde se ejecute:

* Lenguajes de servidor: Se ejecutan en el servidor. Ejemplo: PHP es el más utilizado en servidores web.
* Lenguajes de cliente: Se ejecutan en el cliente. Ejemplo: JavaScript en navegadores web.

**8. Concurrencia.**

Según admitan o no concurrencia de procesos, esto es, la ejecución simultánea de varios procesos lanzados por el programa:

* Lenguajes concurrentes. Ejemplo: Ada.
* Lenguajes no concurrentes. Ejemplo: C.

**9. Interactividad.**

Según la interactividad del programa con el usuario u otros programas:

* Lenguajes orientados a sucesos: El flujo del programa es controlado por la interacción con el usuario o por mensajes de otros programas/sistema operativo, como editores de texto, interfaces gráficos de usuario (GUI) o kernels. Ejemplo: VisualBasic, lenguajes de programación declarativos.
* Lenguajes no orientados a sucesos: El flujo del programa no depende de sucesos exteriores, sino que se conoce de antemano, siendo los procesos batch el ejemplo más claro (actualizaciones de bases de datos, colas de impresión de documentos, entre otros). Ejemplos: Lenguajes de programación imperativos.

**10. Realización visual.**

Según la realización visual o no del programa:

* Lenguajes de programación visual: El programa se realiza moviendo bloques de construcción de programas (objetos visuales) en un interfaz adecuado para ello. No confundir con entornos de programación visual, como Microsoft Visual Studio y sus lenguajes de programación textuales (como Visual C#). Ejemplo: Mindscript.
* Lenguajes de programación textual: El código del programa se realiza escribiéndolo. Ejemplos: C, Java, Lisp.

**11. Determinismo.**

Según se pueda predecir o no el siguiente estado del programa a partir del estado actual:

* Lenguajes deterministas. Ejemplos: Todos los anteriores.
* Lenguajes probabilísticos o no deterministas: Sirven para explorar grandes espacios de búsqueda, (como gramáticas), y en la investigación teórica de hipercomputación. Ejemplo: mutt (generador de texto aleatorio).

**12. Productividad.**

Según se caractericen por tener virtudes útiles o productivas, u oscuras y entretenimiento:

* Lenguajes útiles o productivos: Sus virtudes en cuanto a eficiencia, sencillez, claridad, productividad, etc., motiva que sean utilizados en empresas, administraciones públicas y/o en la enseñanza. Ejemplos: Cualquier lenguaje de uso habitual (C, Java, C++, Lisp, Python, Ruby, entre otros.
* Lenguajes esotéricos o exóticos: Inventados con la intención de ser los más raros, oscuros, difíciles, simples y/o retorcidos de los lenguajes, para diversión y entretenimiento de algunos programadores. Ejemplo: Brainfuck.

## Lenguajes por lugar de ejecución

Los lenguajes de programación para aplicaciones web, dada su naturaleza son ejecutados en sistemas distribuidos, en donde coexisten un computador servidor y uno de tipo cliente como mínimo, por lo que los lenguajes que operan en ambientes web del lado del cliente son HTML, JavaScript, y CSS. Tenemos a PHP, ASP, JSP, Python, Ruby como ejemplos de lenguajes que ejecutan del lado del servidor.

### Lenguaje PHP

PHP es un lenguaje de secuencias de comandos del servidor, y es una herramienta de gran alcance para hacer páginas web dinámicas e interactivas rápidamente.

Ventajas:

* Muy fácil de aprender.
* Se caracteriza por ser un lenguaje muy rápido.
* Soporta en cierta medida la orientación a objeto. Clases y herencia.
* Es un lenguaje multiplataforma: Linux, Windows, entre otros.
* Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
* Capacidad de expandir su potencial utilizando módulos.
* Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
* Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
* Incluye gran cantidad de funciones.
* No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

"PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico." (PHP Document, 2013)

### JavaScript

JavaScript es un lenguaje de programación dinámico, se utiliza con mayor frecuencia como parte de los navegadores web, cuyas implementaciones permiten scripts del lado del cliente para interactuar con el usuario, controlar el navegador, se comunican de forma asíncrona, y altera el contenido del documento que se muestra.

Ventajas:

* Lenguaje seguro y fiable.
* Los script (instrucciones) tienen capacidades limitadas, por razones de seguridad.
* El código Javascript se ejecuta de lado del cliente (navegador web).

### Librería JQuery

jQuery es una libreía de peso ligero para escribir menos y hacer más usando JavaScript. El propósito de jQuery es facilitar el uso de JavaScript estándar en un sitio web. jQuery tiene diversas tareas comunes que requieren muchas líneas de código JavaScript para cumplir, y los envuelve en los métodos que se pueden llamar con una sola línea de código.

jQuery también simplifica mucho las cosas complicadas de JavaScript, como las llamadas AJAX y manipulación DOM.

La biblioteca jQuery permite en un documento web:

* Manipulación HTML
* Manipulación de CSS
* Efectos y animaciones

"jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones (FLV) y agregar interacción con la técnica AJAX a páginas web." (Jquery, 2013)

### JavaScript Object Notation

JSON es la abreviatura de JavaScript Object Notation, y es una forma de almacenar la información de una manera fácil y organizada. En pocas palabras, nos da una colección legible de los datos que podemos acceder de una manera muy lógica. Un ejemplo de su uso es la notación siguiente:

var jason = {

"edad" : "24",

"domicilio" : "Pachuca, Hgo.",

"genero" : "Masculino"

};

### Hojas de estilo de Cascada

Es un lenguaje de estilo que sirve de complemento a los documentos HTML para hacer modificaciones en presentación, es decir, el aspecto de los elementos presentes en un documento HTML.

“CSS abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y muchos otros temas”. (HTML.NET, 2014).

Beneficios de usar CSS:

* Control de la presentación de múltiples documentos desde una única hoja de estilo.
* Control más preciso de la presentación.
* Aplicación de diferentes presentaciones a diferentes tipos de medios (pantalla e impresión).

### Lenguaje HTML

HTML es un lenguaje de marcado; es decir, mediante etiquetas se indica la estructura de un documento, lo que hace posible presentar información en páginas web. Lo que el usuario ve al visualizar una página en Internet es la interpretación que hace el navegador del código HTML.

## Arquitectura

Para el desarrollo de proyectos como el descrito en éste documento, es preciso recurrir al uso y adopción de una metodología de desarrollo de software. Esto permite seguir prácticas de ingeniería de software ya probadas y que funcionan.

### Estructura cliente servidor

El modelo cliente-servidor es una estructura de aplicaciones distribuidas que divide las tareas o las cargas de trabajo entre los proveedores de un recurso o servicio, llamados servidores, y los solicitantes de servicios, denominadas clientes.

A menudo, los clientes y los servidores se comunican a través de una red informática, pero el cliente y el servidor pueden residir en el mismo sistema. Un host de servidor se ejecuta uno o más programas de servidor que comparten sus recursos con los clientes. Un cliente no comparte ninguna de sus recursos, pero solicita contenido del servidor o de la función de servicio. Por lo tanto, los clientes inicien sesiones de comunicación con los servidores que esperan peticiones entrantes.

### Modelo Vista Controlador

También conocido por sus siglas MVC, es un patrón de arquitectura de software para la implementación de interfaces de usuario. Divide una aplicación de software dado en tres partes interconectadas, a fin de separar las representaciones internas de la información, de las formas en que se presenta ésta o que acepte del usuario. Esas partes interconectadas son el modelo, las vistas y los controladores.

El modelo se compone de datos de la aplicación, reglas de negocio, la lógica y funciones. Una vista puede ser cualquier representación de salida de información, como una tabla o un diagrama. En la tercera parte, el controlador, acepta la entrada y la convierte en comandos para el modelo o la vista, es el intermediario entre modelo y vista.

“MVC es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos”. (Desarrolloweb, 2014).

## Frameworks para desarrollo ágil

El desarrollo de aplicaciones por lo general es un proceso tedioso y prolongado, debido a la cantidad de aspectos a considerar, sin embargo es posible desarrollar a partir de proyectos de software existentes.

Un Framework puede ser considerado como la manera de usar uno o varios lenguajes de programación, con el objeto de facilitar su mantenimiento, desarrollo y adopción, es decir, el esqueleto de una aplicación.

### Framework Back-end

Es la labor que compone el acceso a bases de datos, procesamiento y generación de salidas del lado del servidor. El backend se encarga de implementar cosas como MySQL, Postgres, SQL Server o MongoDB. Luego, un lenguaje como PHP o JSP, o frameworks como RoR, Django, Node.JS, Laravel o .NET se conectan a la base de datos.

"En diseño de software es la parte que procesa la entrada desde el front-end" (Laravel, 2013)

### Framework Front-End

Los Frameworks front-end son los encargados de maquetar la estructura del contenido de un documento HTML, codificar el diseño en hojas de estilo (mediante CSS) y agregar la interacción con el usuario (usando JavaScript), esto es la presentación de las aplicaciones. Como ejemplo de Frameworks Front-end están Foundation, Bootstrap, HTML5 Boilerplate y Skeleton.

"En diseño de software el front-end es la parte del software que interactúa con el o los usuarios". (Laravel, 2013)

## Framework Laravel

Laravel es un Framework PHP desarrollado por Taylor Otwell, y que propone una forma de desarrollar aplicaciones web de un modo mucho más ágil. Por ejemplo, en Laravel opcionalmente podemos usar el patrón de diseño MVC (Modelo-Vista-Controlador) tradicional, donde al igual que otros fameworks PHP, el controlador es programado como una clase.

**Ventajas**:

* No tiene requerimientos exóticos para ser ejecutado.
* Dispone de una muy buena y completa documentación en línea.
* Posee una comunidad creciente en el apoyo de este proyecto.

“Desarrollar aplicaciones usando Laravel es muy sencillo, fundamentalmente debido a su expresiva sintaxis y sus generadores de código”. (Valera, 2014).

## Framework Bootstrap

Bootstrap es un conjunto de código HTML, CSS y JavaScript gratuito que permite el desarrollo de interfaces en ambiente web de forma rápida, debido a que el desarrollo no comienza desde nada, sino que el desarrollo parte de un “esqueleto” de sitio web.

### Características

Bootstrap posee características interesantes, es por ello quees integrado en el proyecto, a continuación se enlistan algunas de sus características.

* Bootstrap es compatible con las últimas versiones de los principales navegadores
* Compatibilidad con el diseño web responsivo, es decir, la interfaz de usuario se adapta a la pantalla del dispositivo en cuestión (PC de escritorio, tableta o teléfono móvil).
* Es un proyecto de código libre, por lo que no es necesario cubrir cuotas de licencia por concepto de uso.

## Introducción a las bases de datos

A un conjunto de información relacionada que esté agrupada o estructurada podremos llamarle base de datos, sin embargo al ser parte de un sistema informático una base de datos es una colección organizada de datos. Los datos suelen estar organizados para modelar los aspectos relevantes de la realidad de una manera que apoye los procesos que requieren esta información. En el marco de las bases de datos, se cuenta con diferentes tipos, entre ellas; las relacionales, las orientadas a objetos y las tabulares.

“Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso”. (Dmoz, 2014).

### Base de datos relacionales

Una base de datos relacional es una base de datos que tiene una colección de tablas de elementos de datos, las cuales se describen. Los datos de una sola tabla representan una relación. En las tablas pueden estar definas sus relaciones con las demás. Los tipos de bases de datos relacionales más populares son Oracle, MySQL, PostgreeSQL, SQLite y Microsoft SQL Server.

### MySQL

MySQL es un sistema de gestión de base de datos relacional muy popular, no es de código libre sin embargo, su utilización no tiene costo, tiene la peculiaridad de ser muy compatible con el lenguaje PHP. Es multiplataforma, es decir, puede ser instalado en un PC Windows, unix o Linux o en Mac OS.

## Sistema de información

Un Sistema de Información, es aquél que permite recopilar, administrar y manipular un conjunto de datos que conforman la información necesaria para que los estamentos ejecutivos de una organización puedan realizar una toma de decisiones teniendo la información presente. Como se puede observar en el diagrama I.1 se muestra gráficamente el flujo común de un sistema de información. Nótese que es necesaria la intervención del usuario, que es un ente humano.

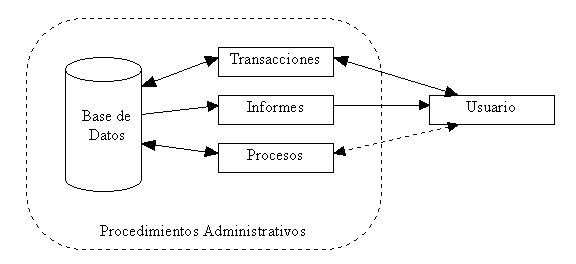


Diagrama I.1. Sentido del flujo de la información en un S.I.

“Un sistema de Información es un conjunto ordenado que permiten manipular toda aquella información necesaria para implementar aspectos específicos de la toma de decisiones.” (José Miguel Santibáñez Allendes, 2014).

# CAPÍTULO II: METODOLOGÍA

## 2.1. Introducción a la metodología

La metodología a emplear para el desarrollo de éste proyecto es Extreme Programming; que también se le puede consultar como XP Programming, programación XP, o Programación Extrema. Según el portal web del XP Programming (Extreme Programming, 2013), ésta metodología impulsa un proyecto de software al permitirle comunicación, simplicidad, retroalimentación, respeto y coraje.

Debido a que cada día emergen recursos y herramientas útiles para acelerar el desarrollo de sistemas de información en ambiente web, la programación extrema resulta interesante debido a que prioriza y da valor a todo aquello que sea sencillo.

“La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo”. (Wesley, 2002).

### 2.1.1. Características

Se tiene por características que la metodología XP, se enfoca en impulsar un proyecto en cinco maneras, y esto es mediante:

* Comunicación: constante entre usuario y desarrollador.
* Simplicidad: En el proyecto, arquitectura sencilla de comprender.
* Retroalimentación: Haciendo pruebas constantes, hacer pruebas por partes.
* Respeto: Valorar cada modificación realizada en el proyecto
* Coraje: para mejorar el proyecto; no importando que ello implique desechar código trabajado anteriormente.

“Extreme Programming improves a software project in five essential ways; communication, simplicity, feedback, respect, and courage.” (Extreme Programming, 2013).

## 2.2. Aplicación de la metodología

Para el presente proyecto de software Programación XP, resulta interesante para aplicar debido a no complica las tareas. La metodología indica la ejecución de trabajo en equipo; sin embargo para el caso de este proyecto, el desarrollo del mismo será de manera individual, los demás criterios de la metodología serán respetados puntualmente.

La metodología XP, refiere la aplicación de cinco rubros importantes y a continuación se explica la forma en que serán aplicadas aquellas cinco rúbricas propuestas por la metodología.

### 2.2.1. Comunicación

Comunicación constante entre usuario y desarrollador. Debido a que no se contará con un usuario o grupo de usuarios definidos a priori, la comunicación sólo será posible con apoyo de recursos electrónicos, (e-mail y formularios) y una vez puesto en producción el sistema informático. Éste punto propuesto por la metodología será aplicado mediante la inclusión de formas de contacto y exhibición del correo electrónico del desarrollador con el objetivo de dar solución a posibles fallos por parte de la aplicación.

### 2.2.2. Simplicidad

Simplicidad en el proyecto, arquitectura sencilla de comprender. Esto se cumplirá al integrar sólo los módulos necesarios a la aplicación y evitar la inclusión de librerías no necesarias, así mismo, la distribución de directorios deberá ser reducida tanto como sea posible.

### 2.2.3. Retroalimentación

Retroalimentación haciendo pruebas constantes, hacer pruebas por partes. Éste punto que indica la metodología XP, sugiere efectuar pruebas en el proyecto de software para detectar oportunamente fallos o vulnerabilidades, con el objeto de evitar fallos en la aplicación una vez puesto éste en producción, Éste punto (especialmente importante, debido a que compromete la funcionalidad del sistema informático) será aplicado gracias a la implementación de pruebas unitarias en el código de la aplicación.

### 2.2.4. Respeto

Valorar cada modificación realizada en el proyecto. Considerar importante cada modificación efectuada ya sea en el código fuente o en la estructura del proyecto, para ello se hará uso de control de versiones mediante la incorporación de Github como herramienta computacional (descrita más adelante) en el proyecto.

### 2.2.5. Coraje

Tener el coraje para mejorar el proyecto; no importando que ello implique desechar código trabajado anteriormente. Pese a que parezca una rúbrica que contradice a la descrita anteriormente, lo que sugiere es atreverse a si, tal vez cambiarlo todo, si esto es más benéfico que perjudicial, al igual que en el punto anterior, la forma de llevarlo a cabo será teniendo un control de versiones del proyecto, por lo que Github nuevamente será la herramienta a utilizar para aplicar éste rubro de la metodología.

# CAPÍTULO III: ESTADO DEL ARTE

## 3.1. Introducción a proyectos alternos

En el mercado podemos encontrar empresas dedicadas al desarrollo de sitios web, sin embargo; se encuentran inconvenientes como lo son; publicidad, diseños similares entre sí, algunas como Jimdoo*®* resultan ser de mala calidad, tomando como referencia, los principios de usabilidad web, o reglas de oro sobre usabilidad de Jakob Nielsen:

* En Internet, el usuario es el que manda
* En Internet la calidad se basa en la rapidez y la fiabilidad.
* Seguridad.
* La confianza es algo que cuesta mucho ganar y se pierde con un mal enlace.
* Si quieres hacer una página decente, simplifica, reduce, optimiza.
* Pon las conclusiones al principio.
* No hagas perder el tiempo a la gente con cosas que no necesitan.
* Buenos contenidos.

Según Jakob Nielsen, autor del libro Usabilidad. Diseño de páginas Web*,* un sitio web con usabilidad…

“Es aquél que muestra todo de una forma clara y sencilla de entender por el usuario. Aunque es imposible crear un sitio que sea claro y eficiente para cada usuario, el diseñador debe esforzarse para mostrar las cosas tan claramente como sea posible, de tal modo que reduzca al mínimo cualquier aspecto que pueda ser confuso”. (Nielsen, 2000).

Jakob Nielsen, sugiere diseñar sitios web pensando en el usuario, en lograr que la interfaz denote sencillez y claridad, lo anterior por que deben reducirse las confusiones y todo debe ser tan claro como sea posible. Optar por minimalismo en el diseño, es una muy buena opción, claro está; siempre pensando en el usuario, debemos recordar que el presente sistema, deberá ser operado por personas que cuenten con pocos o nulos conocimientos sobre concepción y diseño web.

Actualmente existen empresas privadas dedicadas a la creación de sitios web, algunas como *WIX*® (Wix, 2013) ofrecen herramientas fáciles de usar, sin embargo los diseños propuestos son basados en imágenes, agregan publicidad y hasta cierto punto es necesario saber sobre diseño web para obtener buenos resultados. Por otra parte también está 1&1 (1&1 México, 2013), que es una empresa Alemana con un buen servicio, pero los diseños web con que cuenta son basados en plantillas CSS monótonas, y no hay aprovechamiento de recursos Front End recientes (transformadas CSS, JQuery, entre otros.).

Como empresa con soluciones alternativas a las anteriores y que sea orientada a PyMES es Jimdoo con *Conecta tu negocio* que resulta de la iniciativa de Google, el propio Jimdoo, Banamex, Consejo de la comunicación, Instituto PyME y Universidad Virtual del Tecnológico de Monterrey y que cuenta con la colaboración de la Secretaría de Economía y México Emprende, lo anterior según su portal web (Conecta tu negocio, 2013).

## ESTADO DEL ARTE

Por lo general para las microempresas hidalguenses y mexicanas, el presupuesto disponible para tecnologías de información con que cuentan es bajo, más aún lo es para tener un sitio web. El sistema de información tiene perfil para PyMES, debido a que es lo que más abunda en el país, el mercado en potencia, en tanto es grande.

“Las Pequeñas y Medianas Empresas en México ofrecen características y necesidades muy particulares. Por un lado, se constituyen como parte fundamental del andamiaje económico, ya que representan el 99.7 por ciento de los aproximadamente 4 millones de empresas en México, las cuales generan el 52 por ciento del Producto Interno Bruto Nacional y además contribuyen con la creación del 71.9 por ciento de los empleos. Por otra parte, las Pequeñas y Medianas Empresas tienen un alto grado de mortalidad, el 50 por ciento de las empresas quiebran con tan solo un año de actividad y el 90 por ciento de las empresas mueren antes de cumplir los 5 años debido en gran parte a la falta de financiamiento para desarrollarse y expandirse.” (Perales, 2013)

La falta de financiamiento de las PyMES para desarrollarse y expandirse denota un bajo presupuesto con que cuentan, por lo que es importante el desarrollo de una plataforma de bajo costo, y que se adapte al presupuesto con que cuenta el promedio de las PyMES.

Además de lo anterior, cabe mencionar que la persona que administra la organización generalmente no cuenta con los conocimientos técnicos necesarios para poder crear un sitio web de calidad. El problema al que se enfrentan los emprendedores de nuestro estado y más aún; nuestro país, es el bajo presupuesto o en su defecto los pocos conocimientos que son requeridos para tener un sitio web.

Consultando en el mercado local, está como ejemplo SICOMWEB, quienes en su portal web dicen:

“Ofrecemos el diseño de páginas web económicas” (Sicom Web, 2013)

Para el diseño de sitios web, según el mismo portal; el costo oscila entre los $1,500.00 y $3,000.00 pesos mexicanos anuales (Consultado el día 19 de Octubre del 2013). El precio es bien justificado debido a que el trabajo se hace de forma manual, además de que requiere de personal especializado con conocimientos refinados sobre concepción y diseño web. Debido al costo elevado, resulta asequible sólo para aquellos microempresarios que destacan o que tienen alto volumen de ventas como para que estén dispuestos a invertir en un sitio web corporativo.

También tenemos alternativas automatizadas basadas en plantillas como por ejemplo *Google Sites*®, *Wix*®, *Jimdoo*® y 1*&*1®. Google Sites® es gratuita y Jimdoo® para PyMES también lo es, pero ésta última sólo durante el primer año de contratación. El problema radica en que estas alternativas siguen siendo complejas de entender para el promedio de emprendedores y microempresarios hidalguenses, esto debido a su complejidad, en algunos casos es preciso conocer sobre formatos de imágenes digitales. Para el caso de servicios no gratuitos, es necesario el hacer pagos electrónicos, y debido a factores como la falta de infraestructura, conocimientos técnicos y costumbre en manejo de dinero por internet; lograr que una persona interesada en la gestión de un sitio web en México (y más aún en Hidalgo) sea conducido desde el registro hasta el pago electrónico; resulta difícil.

</>

# CAPÍTULO IV: HERRAMIENTAS COMPUTACIONALES

## Herramientas a utilizar en el proyecto

A continuación se enlistas las herramientas computacionales a utilizar en el proyecto, cabe mencionar que por la naturaleza del proyecto en ser orientado a web, se recurre a tecnologías soportadas desde cualquier navegador web moderno. Para el caso de las tecnologías que son ejecutadas del lado del servidor para el usuario no deben significar inconveniente alguno, el proceso es encpasulado.

**AJAX**

JavaScript Asíncrono con XML, es una técnica aplicada a páginas web que permite inclusión de contenido interactivo.

**CSS**:

Son hojas de estilo en cascada y sirven para añadir propiedades de estilo, como lo son forma, tamaño, color, la presentación del contenido de una página web.

**HTML:**

Lenguaje de etiquetas que contiene información.

**JQuery:**

**MySQL:**

"MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario" (MySQL, 2013)

**PHP**

**XML:**

"XML, siglas en inglés de eXtensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible." (World Wide Web Consortium, 2013)

Marco teórico:

Para el diseño de sitios web es requerido trabajar con páginas web, hojas de estilo y técnicas Ajax. Lo anterior puede realizarse de forma manual; es decir, codificando las instrucciones en su respectivo lenguaje (HTML, CSS o JavaScript), sin embargo la tarea puede automatizarse. Actualmente en tentupagina.com se trabaja el diseño de sitios web de forma manual, el servicio es orientado a PyMES, y como necesidad se tiene la de crear un panel de administración para cada sitio web, de forma tal que desde el mismo, el cliente pueda gestionar el contenido y diseño su sitio web, sin requerir de la intervención del diseñador o desarrollador web.

Cabe señalar que para el caso de PyMES y emprendedores, la persona cliente encargada de administrar el sitio web, por lo general cuenta con conocimientos básicos o nulos sobre diseño web, por lo que es importante el desarrollo de una plataforma que sea tan fácil, intuitiva y simple de usar como el correo electrónico, por poner un ejemplo. Aunado a lo anterior debe contemplarse el desarrollo de una solución viable y económica, lo anterior con el propósito de ofrecer un servicio que tenga un costo que esté al alcance de las PyMES y emprendedores.

</>

# CAPÍTULO V: ANÁLISIS Y DISEÑO DEL SISTEMA

Justificación

El desarrollo de una plataforma de apoyo para la creación de sitios web corporativos orientados a PyMES requiere que para ésta sea fácil incorporar tecnologías

contemporáneas, y esté preparada para dar soporte a tecnologías futuras. Por ello la importancia de usar *Programación XP* como metodología de desarrollo.

Se debe lograr que la plataforma y los sitios web creados con la misma sean sencillos y fáciles de usar, es aquí donde encontramos la importancia del uso de tecnologías *HTML*, *CSS* y *Jquery* para el *Front End*.

Se hace el uso de PHP, MySQL y XML en Back End debido a que son recursos libres, económicos y muy compatibles con infraestructuras existentes, servicios de hosting en el mercado y servidores comerciales.

Para el desarrollo se opta por el uso de LAMPP, terminal de comandos, navegador Chromium y Gedit por ser herramientas ligeras y de uso libre.

## Metas

* Terminar el desarrollo del sistema en menos de un año.
* Desarrollar una zona de registro sencilla, que requiera de pocos campos
* Dar soporte a múltiples sitios web (mínimo 100)
* Limitar presupuesto a sólo gastar en dominios, usar proveedores de hospedaje gratuito.
* Desarrollar un instalador de aplicación (Script creador de Base de Datos)

## Hipótesis

Mediante el empleo de tecnologías de información y recursos web; usando una arquitectura cliente-servidor, es posible el desarrollo de una aplicación basada en web que permita a usuarios sin conocimientos técnicos crear, modificar y publicar sitios web de tipo corporativo orientados a PyMES.

## Alcances y limitaciones

El sistema sólo diseñará sitios web con las siguientes características:

* Hasta 5 páginas por sitio web.
* Diseño web no personalizable.
* Sólo diseño corporativo, sitios web informativos.
* No será posible la modificación del sitio por medio de código fuente.
* Las imágenes y/o fotografías, así como el contenido del sitio web, será proporcionado por el usuario.

Objetivos especificos

Desarrollar una aplicación e implantar el sistema una vez terminado, así como también el cumplir con las siguientes características tanto para el sistema como para los sitios web construidos con el mismo.

* Zona de registro (Rápido) y acceso.
* Construcción fácil y rápida a partir plantilla o a partir de maquetación y colores.
* Procesamiento de la información proporcionada por el usuario para su inyección en el sitio web.
* Posibilidad de impresión de formas de pago para el caso de solicitud de propio dominio, solicitud de más espacio de hospedaje o adición de módulos.
* Fácil adopción de tecnologías web emergentes.
* Unidad de pruebas y respaldo.
* Hospedaje web y dirección URL para consulta.
* Uso de tecnologías de vanguardia en cuanto a diseño web, sitios web vistosos, llamativos y profesionales.
* Adopción de técnicas SEO.
* Unidad de pruebas y respaldo.

## Propuesta de solución

Es preciso el desarrollo de un sistema de información orientado a la web, que permita la construcción de sitios para internet y que sea tan fácil de usar, se pretende que la plataforma sea operada por los propios emprendedores y dueños de PyMES, que no necesite de una persona especializada para su completa operación.

Debido a que en la web surgen cambios importantes y de manera constante, se opta por la adopción de *programación extrema*, como metodología de desarrollo de ingeniería de software. Según el sitio web oficial de *Programación XP,* ésta metodología mejora un proyecto de software de 5 maneras:

* Comunicación: Constante entre clientes y equipo de desarrolladores.
* Simplicidad: Diseño limpio y claro.
* Retroalimentación: Al hacer pruebas con el software desde el principio
* Respeto: A cada contribución al proyecto, valorar éxitos
* Coraje: Para cambiar requerimientos y tecnología.

Es importante el lograr el desarrollo de una plataforma que sea extensible; que permita añadir elementos emergentes sin afectar el resto de la funcionalidad de la aplicación.

Las tecnologías a utilizar para el desarrollo de la aplicación son:

* **Front-end:** HTML, CSS, JQuery, JavaScript
* **Back-end:** PHP, MySQL, XML

Se pretende tentativamente emplear algún *Framework* *MVC* *PHP*, como Zend™, Laravel o Drupal por ser marcos de trabajo que agilizan el desarrollo de aplicaciones (Back-end).También se contempla el utilizar *Frameworks* para el *Front-End*, como Bootstrap de Twitter™ o Foundation.

Las herramientas de apoyo para el desarrollo son: navegador Chromium, *LAMPP*, terminal de comandos, y *Gedit*. No se hará uso de alguna IDE como *Aptana* o *Eclipse*, esto por motivos de rendimiento y libertad de programación.

Para la implantación del proyecto es requerido del servicio de *Hosting*, por lo que se manejará uno de tipo Linux compartido. La plataforma será orientada a la web, esto para facilitar su acceso desde cualquier dispositivo.

### Mapas de navegación

Ilustración V.1: Estructura de los sitios web autogenerados

Ilustración V.2: Estructura del panel de administración del sitio web

Ilustración V.3: Estructura de la página de registro para acceder al servicio

### Casos de uso

Zona de registro

Registro

Cancelar registro

Consultar políticas

Usuario NO registrado

Usuario NO registrado

Administrador sitio web (usuario REGISTRADO)

Sitio Web

Consultar sitio

Panel de Administración sitio web

Consultar políticas

Publicar sitio web

Modificar diseño/contenido sitio web

Eliminar cuenta

Ilustración V.4: Diagrama de casos de uso de la aplicación web de tentupagina.com

## Prototipo

La aplicación es operada utilizando un panel de administración, desde éste es posible la edición del diseño y contenido del sitio web, incluye las herramientas necesarias para administrar el sitio de forma rápida y sencilla así como también sugerencias de uso, para que con ello el usuario no requiera de leer a detalle el manual de usuario.

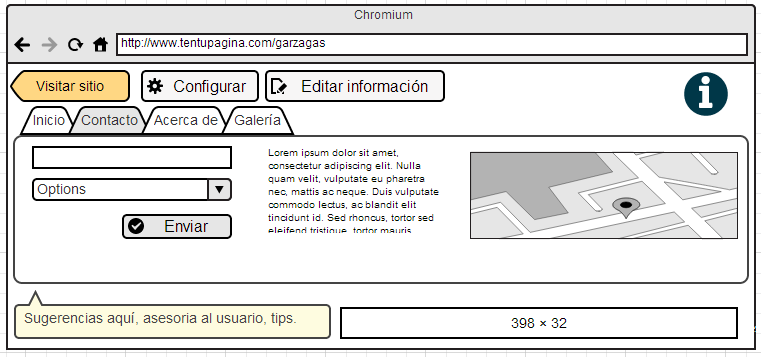


Ilustración V.5: Boceto Panel de administración sitio web. Pantalla con panel de control de página web.

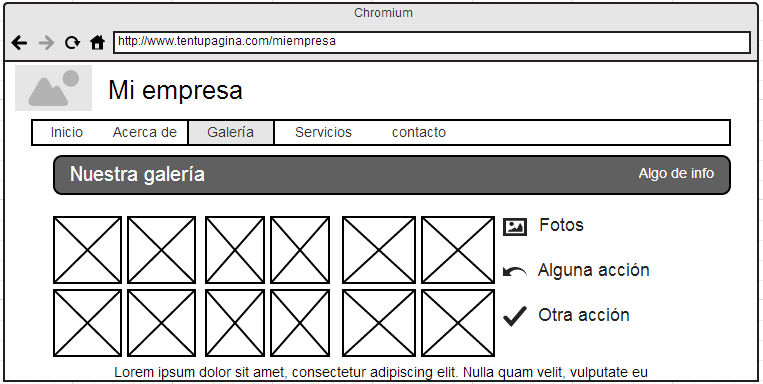


Ilustración V.6: Boceto de sitio web. Aspecto de sitio web generado mediante la aplicación, como se puede observar es un sitio web de tipo corporativo informativo.

</>

# CAPÍTULO VI: DESARROLLO DEL SISTEMA

## Desarrollo

El sistema de información se escribe en lenguajes web los cuales son:

PHP Es un lenguaje que se ejecuta del lado del servidor, es decir; debe coexistir una arquitectura de tipo cliente – servidor en la aplicación, por tanto debe haber un usuario con dispositivo cliente que hace peticiones al dispositivo servidor, una vez hecha la petición…

PHP fue originalmente desarrollado por el danés Rasmus Lerdorf groenlandés, y fue desarrollado posteriormente como código abierto. PHP no es un estándar web adecuada -, sino una tecnología de código abierto. PHP es ni lenguaje de programación real - pero PHP permite utilizar los llamados scripts en sus documentos.

Para describir lo que una página PHP es, se podría decir que se trata de un archivo con la extensión. Php que contiene una combinación de etiquetas HTML y secuencias de comandos que se ejecutan en un servidor web.

De lado del servidor:

* Json
* XML
* PHP

De lado del cliente:

* CSS
* JavaScript
* JQuery
* HTML

Documentación técnica automática del proyecto

* MD (Archivos Markdown)

Para el control de versiones se maneja Github. Se tienen tres ramas de proyecto:

* **Master**: Sistema de información en producción
* **Beta**: Sistema entregable pero propenso a errores
* **Developer:** (Desarrollo): Sistema en desarrollo

El proyecto tiene repositorio en la nube, es decir; puede ser monitoreado, clonado y descargado desde la siguiente URL: <https://github.com/adlnetworks/ttp-app/>

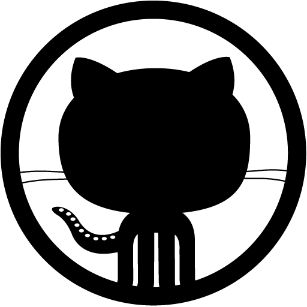
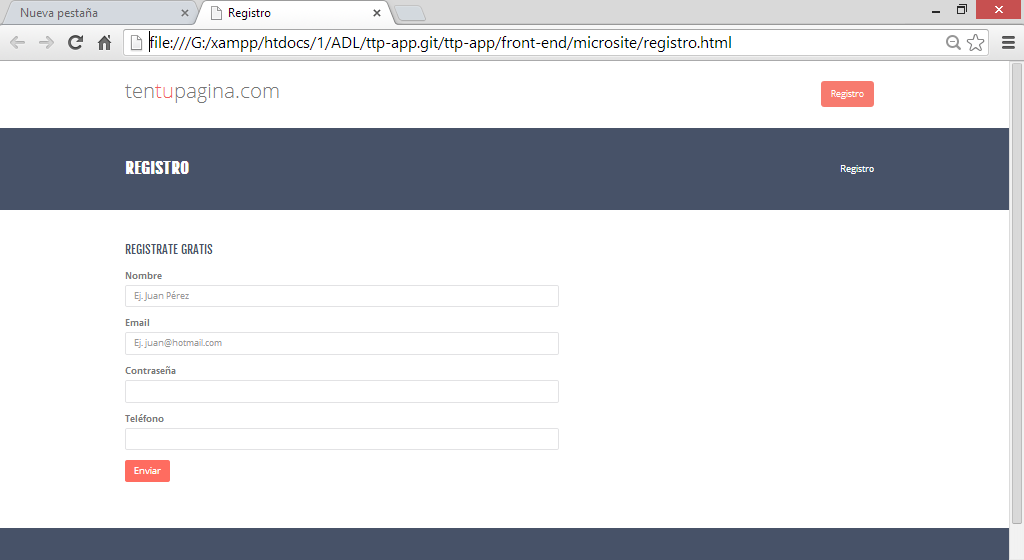
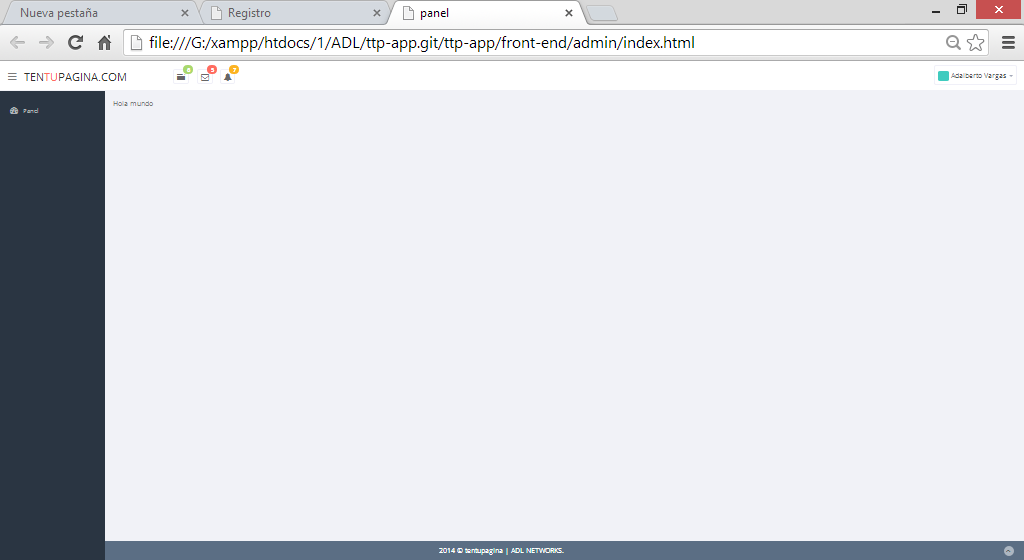


Ilustración VI.1: Github logo, el usar control de versiones es una buena práctica en desarrollo de software

### Creación de formulario de registro



### Comienzo de desarrollo de panel



### Definición de estilos CSS (20 líneas de 7098)

article,

aside,

details,

figcaption,

figure,

footer,

header,

hgroup,

main,

nav,

section,

summary {

display: block;

}

html {

font-family: sans-serif;

-webkit-text-size-adjust: 100%;

-ms-text-size-adjust: 100%;

}

### Responsive CSS (20 de 415 líneas de código)

@media (min-width: 980px) {

/\*-----\*/

.custom-bar-chart {

margin-bottom: 40px;

}

}

@media (min-width: 768px) and (max-width: 979px) {

.mail-box .sm-side {

width: 30%;

}

.mail-box .lg-side {

width: 70%;

}

/\*-----\*/

.custom-bar-chart {

margin-bottom: 40px;

}

### Integración de repositorio de librerias JavaScript

jquery-ui-1.9.2.custom.min.js

jquery.sparkline.js

jquery.sparkline-11.js

jquery-1.8.3.min.js

jquery.js

jquery.nicescroll.js

bootstrap.js

gmaps.js

ga.js

bootstrap.min.js

owl.carousel.js

jquery.validate.min.js

flot-chart.js

additional-methods.min.js

jquery.tagsinput.js

respond.src.js

jquery.stepy.js

modernizr.custom.js

bootstrap-switch.js

form-component.js

jquery.dcjqaccordion.2.7.js

editable-table.js

gmaps-scripts.js

advanced-form-components.js

hover-dropdown.js

gritter.js

jquery.cookie.js

sparkline-chart.js

respond.min.js

external-dragging-calendar.js

jquery.dcjqaccordion.2.7.min.js

dynamic-table.js

all-chartjs.js

morris-script.js

common-scripts.js

sliders.js

form-validation-script.js

form-image-crop.js

jquery.scrollTo.min.js

html5shiv.js

count.js

jquery.customSelect.min.js

toucheffects.js

nestable.js

jquery.ui.touch-punch.min.js

easy-pie-chart.js

tasks.js

### Generación de vista del panel (20 de 146 líneas)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta name="description" content="">

<meta name="author" content="adl">

<meta name="keyword" content="">

<link rel="shortcut icon" href="img/favicon.png">

<title>panel</title>

<!-- Bootstrap core CSS -->

<link href="css/bootstrap.min.css" rel="stylesheet">

<link href="css/bootstrap-reset.css" rel="stylesheet">

<!--external css-->

<link href="assets/font-awesome/css/font-awesome.css" rel="stylesheet" />

<!-- Custom styles for this template -->

<link href="css/style.css" rel="stylesheet">

<link href="css/style-responsive.css" rel="stylesheet" />

### Maquetado del front de panel

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>

<?php include('includes/meta-header.php'); ?>

</head>

<body>

<div id="wrapper">

<?php include('includes/header.php'); ?>

<?php include('includes/nav.php'); ?>

<div id="content">

<?php include('content/c-inicio.php'); ?>

</div> <!-- end #content -->

<?php include('includes/sidebar.php'); ?>

<?php include('includes/footer.php'); ?>

</div> <!-- End #wrapper -->

</body>

</html>

### Edición de Json RESTful de 20 de 177 líneas

$('#my\_multi\_select1').multiSelect();

$('#my\_multi\_select2').multiSelect({

selectableOptgroup: true

});

$('#my\_multi\_select3').multiSelect({

selectableHeader: "<input type='text' class='form-control search-input' autocomplete='off' placeholder='search...'>",

selectionHeader: "<input type='text' class='form-control search-input' autocomplete='off' placeholder='search...'>",

afterInit: function (ms) {

var that = this,

$selectableSearch = that.$selectableUl.prev(),

$selectionSearch = that.$selectionUl.prev(),

selectableSearchString = '#' + that.$container.attr('id') + ' .ms-elem-selectable:not(.ms-selected)',

selectionSearchString = '#' + that.$container.attr('id') + ' .ms-elem-selection.ms-selected';

that.qs1 = $selectableSearch.quicksearch(selectableSearchString)

### Conexión de Json con admin (18 de 95 líneas)

<?php

class Application

{

/\*\* @var null The controller \*/

private $url\_controller = null;

/\*\* @var null The method (of the above controller),\*/

private $url\_action = null;

/\*\* @var null Parameter one \*/

private $url\_parameter\_1 = null;

/\*\* @var null Parameter two \*/

private $url\_parameter\_2 = null;

/\*\* @var null Parameter three \*/

private $url\_parameter\_3 = null;

</>

# CAPÍTULO VII: PRUEBAS DEL SISTEMA

</>

# Conclusiones

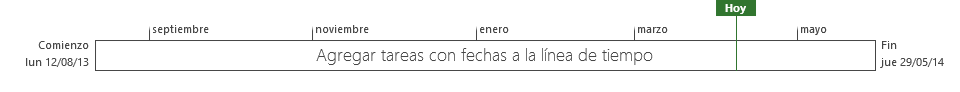
Los resultados de la investigación indican un área de oportunidad en potencia derivado de la necesidad de algunas PyMES de tener presencia en internet, y que es posible satisfacer tal necesidad mediante el desarrollo de una aplicación web. Se pretende lograr que la aplicación sea sencilla, fácil de utilizar y de bajo costo.

Con los resultados obtenidos es posible asegurar el desarrollo de la aplicación mediante tecnologías para web, debido a la naturaleza de la misma, la metodología a emplear, se adapta de forma adecuada de modo tal que son pocos los ajustes necesarios, como aporte tecnológico se tiene a la aplicación que se espera pueda resultar de utilidad para comunidad que requiera de algún sitio web.

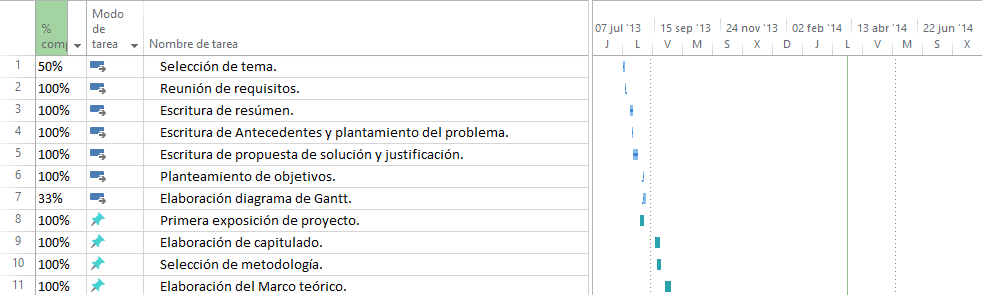
El surgimiento reciente de nuevas necesidades como visualizar sitios web desde teléfonos inteligentes, son creadas nuevas tecnologías, tales como el diseño adaptativo o responsivo; que permite la visualización de sitios web; adaptada al tamaño de pantalla de algún dispositivo móvil; ya sea tableta o teléfono. Derivado de lo anterior queda abierta la posibilidad de ofrecer sitios web con diseño responsivo o más aún ofertar aplicaciones basadas en web para dispositivos móviles.

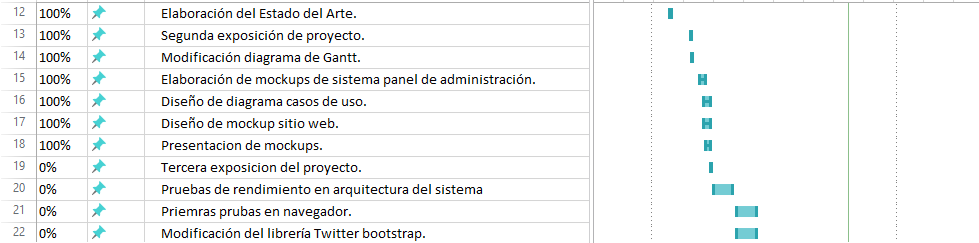
# Anexo I: Cronograma de actividades

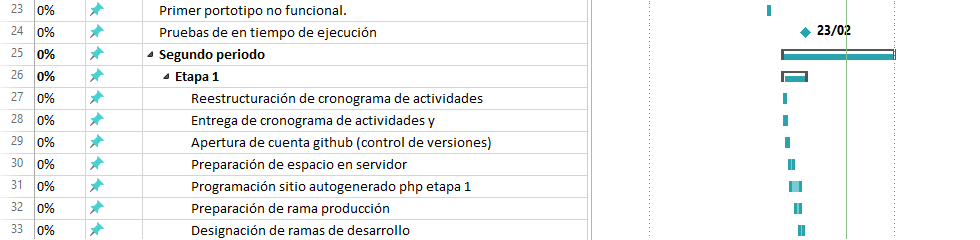
Gráfica general

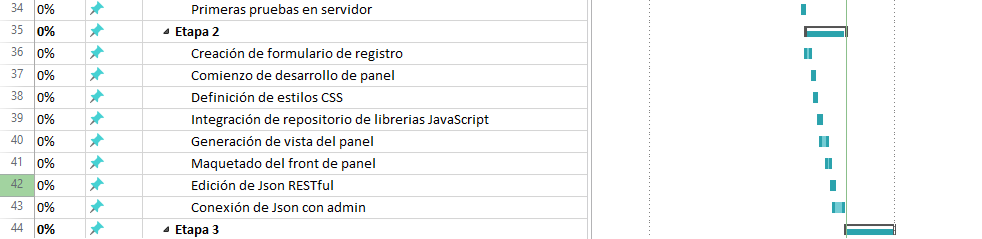


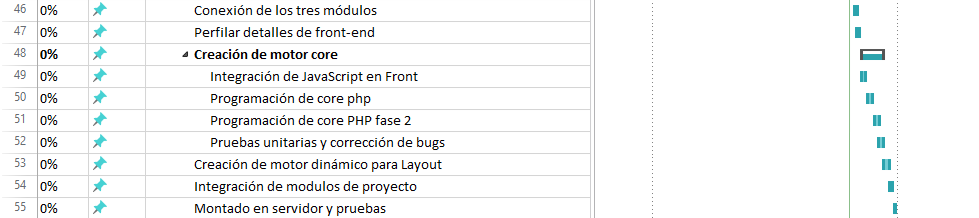
Tareas











Anexo II: Diapositivas

Glosario

Es aquí en dónde se mencionan los principales términos a manejar, y que son importantes para el desarrollo de dela aplicación web tentupagina.com.

</>

# Referencias

Referencias

*1&1 México.* (21 de 08 de 2013). Obtenido de 1&1 México: http://www.1and1.mx/nuevas-extensiones-dominios?linkOrigin=&linkId=ct.stage.img.domain.reservation&stage=ntld

*Conecta tu negocio.* (27 de 08 de 2013). Obtenido de Conecta tu negocio: http://www.conectatunegocio.com.mx/

Extreme Programming. (20 de 10 de 2013). *Extreme Programming*. Obtenido de A gentle introduction: http://www.extremeprogramming.org/

Hernández Perales Norma Alicia, V. M. (23 de 10 de 2013). La Pequeña y Mediana Empresa de México, Datos Relevantes y sus Prácticas de Gobierno Corporativo y Administración de Riesgos. Nuevo León, Monterrey, México.

Jquery. (06 de 09 de 2013). *Jquery.* Obtenido de Introducción: http://es.jquery.org

*Laravel.* (06 de 08 de 2013). Obtenido de Front Vs Back end: http://laravel.com/Front-end\_y\_back-end

*MySQL.* (06 de 09 de 2013). Obtenido de Acerca de: http://oracle.com/mysql

Nielsen, J. (2000). *Usabilidad. Diseño de páginas Web.* Prentice Hall.

*PHP Document.* (06 de 09 de 2013). Obtenido de PHP Empower: http://php.net

Sicom Web. (19 de 10 de 2013). *Sicom Web*. Obtenido de Inicio: http://www.sicomweb.com.mx/index.html

Wesley, A. (02 de 09 de 2002). *La programacion extrema en la practica.* Pearson. Obtenido de wikipedia Español: http://es.wikipedia.org/wiki/Extreme\_Programming

*Wix.* (20 de 08 de 2013). Obtenido de wix: http://es.wix.com/sitiowebgratis/es400?utm\_source=google&utm\_medium=cpc&utm\_campaign=bi\_wix\_sp^wix&experiment\_id=wix^e^26519109183^1t1&gclid=CN6w65nUnbkCFUxo7AodFBwAeg

*World Wide Web Consortium.* (06 de 09 de 2013). Obtenido de XML: http://w3c.org/Xml