

SUBSECRETARÍA DE EDUCACIÓN SUPERIOR
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR
TECNOLÓGICA
INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ



SECRETARÍA DE
EDUCACIÓN PÚBLICA



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA

TRABAJO PROFESIONAL

COMO REQUISITO PARA OBTENER EL TITULO DE:

INGENIERO EN SISTEMAS COMPUTACIONALES

QUE PRESENTA:

ELMER EDER AGUILAR ORDOÑEZ

CON EL TEMA:

“VENTAJAS DEL USO DE FRAMEWORK PARA APLICACIONES WEB.”

MEDIANTE:

OPCION I
(TESIS PROFESIONAL)

TUXTLA GUTIERREZ, CHIAPAS

ABRIL 2016

SEP

SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Tuxtla Gutiérrez

"2015, Año del Generalísimo José María Morelos Y Pavón"

DIRECCIÓN
SUBDIRECCIÓN ACADÉMICA
DIVISIÓN DE ESTUDIOS PROFESIONALES
Tuxtla Gutiérrez, Chiapas 02 de marzo del 2016

OFICIO NUM. DEP-CT-15-2016

C. ELMER EDER AGUILAR ORDOÑEZ

PASANTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES
EGRESADO DEL INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ.
P R E S E N T E.

Habiendo recibido la comunicación de su trabajo profesional por parte de los CC. M.C. JORGE OCTAVIO GUZMAN SANCHEZ, M.C. WALTER TORRES ROBLEDO Y M.C. GALINDO BELIZARIO NANGO SOLÍS en el sentido que se encuentra satisfactorio el contenido del mismo como prueba escrita, **AUTORIZO** a Usted a que se proceda a la impresión del mencionado Trabajo denominado:

" VENTAJAS DEL USO DE FRAMEWORK PARA APLICACIONES WEB."

Registrado mediante la opción:
I (TESIS PROFESIONAL)

ATENTAMENTE
"CIENCIA Y TECNOLOGÍA CON SENTIDO HUMANO"

ING. JUAN JOSÉ ARREOLA ORDAZ
JEFE DEL DEPARTAMENTO DE LA DIVISION DE
ESTUDIOS PROFESIONALES

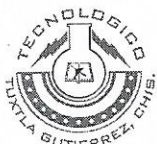
Vo. Bo.

M. en C. JOSÉ LUIS MÉNDEZ NAVARRO
DIRECTOR

C.c.p.- Departamento de Servicios Escolares
C.c.p.- Expediente
I'JLMN/I'JJAQ/I'eeam



Secretaría de Educ. Pública
Instituto Tecnológico
de Tuxtla Gutiérrez,
Div. de Est. Profesionales



Carretera Panamericana Km. 1080, C.P. 29050, Apartado Postal 599
Tuxtla Gutiérrez, Chiapas; Tels. (961) 61 54285, 61 50461
www.ittg.edu.mx



ASDC 942
Fecha de Emisión: 2015-03-02
Fecha de Terminación: 2015-03-02

Agradecimientos

Primero quiero agradecer a Dios por la vida que me ha regalado y haberme permitido llegar hasta donde hoy me ha permitido llegar.

A mi familia por su apoyo. En especial a mis papas que han confiado en mí y me han apoyado en todo este trayecto sin ustedes nada de esto fuera posible los quiero mucho.

Gracias a mi asesor que me apoyo en el trayecto de la tesis enseñándome durante este proceso y su disposición al resolver mis dudas.

Gracias a todos...

Resumen

El presente documento de investigación presenta la importancia del uso y manejo de Frameworks de desarrollo para Aplicaciones Web en el cual se hace un análisis de estos.

Aquí se hace mención del cómo se podría trabajar con y sin un Framework. Haciendo mención de algunas ventajas que se obtienen al desarrollar con ellos, como, por ejemplo:

- Reutilización de código.
- Ahorro de tiempo en el desarrollo.
- Ayuda de librerías.
- Programación Orientada a Objetos.
- Facilidades en el manejo de servidores y base de datos.

Entre otras.

Índice

Capítulo I: Introducción	1
1.1 Antecedentes	1
1.2 Planteamiento del problema	2
1.4 Objetivos	3
1.4.1 Objetivo general	3
1.4.2 Objetivos específicos	3
1.5 Hipótesis	4
1.6 Justificación	5
1.7 Alcances y limitaciones	6
1.7.1 Alcances	6
1.7.2 Limitaciones	6
1.8 Estado del arte	7
1.8.1 Sistemas desarrollados en la Ingeniería de Sistemas Computacionales del ITTG.	7
Capítulo II: Marco teórico	12
2.1 Marco teórico técnico conceptual.	12
Capítulo III. Análisis de Framework	17
3.1. Ruby On Rails	17
3.2. CodeIgniter	19
3.3. Django	22
3.4. Zend Framework	24
3.5. Bootstrap: Framework para CSS	29
3.6 Blueprint: Framework de CSS	31
3.7. ASP.NET	33
Capítulo IV Análisis al desarrollar una Aplicación Web	36
4.1 Aplicaciones web sin usar un Framework	36
4.2 Aplicaciones web usando Framework	37
Capítulo V Propuesta	39
Conclusiones	40
Apéndice A	43
Glosario	48
Referencias	50

Índice de Imágenes

<i>Ilustración 1: Interfaz de Sistema para la Administración y Publicación en Línea de Calificaciones de la Escuela de Terapia Física del DIF Chiapas.</i>	7
<i>Ilustración 2: Interfaz de Sistema Web de Inventario de la Intranet de CFE de la Zona de Distribución de Tuxtla.</i>	8
<i>Ilustración 3: Interfaz de Sistema de Administración de Servicios Catastrales.</i>	9
<i>Ilustración 4: Interfaz Desarrollo de un Sistema de Información en la Coordinación de Agenda Logística del Estado de Chiapas (Módulo Viáticos).</i>	10
<i>Ilustración 5: Interfaz de Sistema de Control de Inventarios de Reactivos que se Surtirán en el PTNDIPAB.</i>	11
<i>Ilustración 6: Representación del funcionamiento del MVC.</i>	14
<i>Ilustración 7: Ruby On Rails.</i>	17
<i>Ilustración 8: Componente de Rails y MVC.</i>	17
<i>Ilustración 9: Servidor de datos Cliente / Servidor.</i>	18
<i>Ilustración 10: Logotipo de CodeIgniter.</i>	19
<i>Ilustración 11: Diagrama de Flujo de CodeIgniter.</i>	19
<i>Ilustración 12: Estructura de carpetas del Framework CodeIgniter.</i>	20
<i>Ilustración 13: Logotipo de Framework django.</i>	22
<i>Ilustración 14: Funcionamiento de MPC Django.</i>	22
<i>Ilustración 15: Logotipo de Framework Zend Framework.</i>	24
<i>Ilustración 16: Estructura básica de una aplicación con Zend Framework.</i>	25
<i>Ilustración 17: Estructura modular básica para ZF.</i>	26
<i>Ilustración 18: Logotipo de Bootstrap.</i>	29
<i>Ilustración 19: Tipos de dispositivos al que se adecuan Bootstrap.</i>	29
<i>Ilustración 20: Distribución de los archivos de Bootstrap.</i>	30
<i>Ilustración 21: Estructura básica de Bootstrap.</i>	30
<i>Ilustración 22: Logotipo de Blueprint.</i>	31
<i>Ilustración 23: Logotipo de ASP.NET.</i>	33
<i>Ilustración 24: Arquitectura MVC de ASP.NET.</i>	34
<i>Ilustración 25: Organización de los archivos de las Aplicaciones Web.</i>	36

Capítulo I: Introducción

El presente trabajo tiene como propósito servir como un marco de referencia para todos aquellos estudiantes que estén interesados en realizar proyectos en la cual requieran la programación y el diseño de Aplicaciones Web.

Uno de los objetivos es demostrarles a los alumnos lo favorable que es manejar un Framework para la programación y diseño de una Aplicación Web, ya que se mencionarán varios de ellos, para así beneficiarse en el ahorro tiempo, escritura de código fuente, además de la seguridad y confiabilidad para el desarrollo.

1.1 Antecedentes

Ingeniería en Sistemas Computacionales es una carrera que forma parte de la oferta educativa del Instituto Tecnológico de Tuxtla Gutiérrez, ésta ha ido evolucionando con el paso del tiempo para garantizar profesionistas con un nivel alto de conocimientos en las diversas áreas a las que se puede dedicar un egresado.

En la medida en que los estudiantes de esta carrera aprenden los trabajos se realizan de una forma desordenada, artesanal, con funciones y clases incompletas o un programa simple que realice la función que se planteó en clases, dando como resultado un trabajo que con el tiempo será difícil de entender y no se podrá seguir desarrollando en un futuro.

Existen muchos proyectos de Aplicaciones Web desarrollados por los estudiantes de la Ingeniería en Sistemas Computacionales, los cuales son presentados como proyectos de residencia o titulación integrada cada año en el departamento de sistemas y computación y en los que este conocimiento podría ayudar.

1.2 Planteamiento del problema

La mayoría de los estudiantes de Ingeniería en Sistemas Computacionales, al realizar proyectos de Aplicaciones Web para residencia se encuentra con las complicaciones al momento de hacer el desarrollo, existen la falta de conocimientos de ciertas reglas de diseño y de una programación ordenada.

El desarrollo de proyectos sin conciencia de algún estándar, política o sin lineamientos nos da la certeza de que el trabajo de residencia, forme parte de una programación desordenado en el cual se es difícil laborar. Inclusive hacerle mejoras para el beneficio de la aplicación y en un futuro lo cual también permite el retraso.

Citando a los proyectos desarrollados para residencia o titulación integrada que son presentados cada año para su evaluación y acreditación del estudiante, encontramos muchos proyectos en la cual son basados en Aplicaciones Web que solicitan diferentes empresas de algún tipo de servicio.

Muchos de los alumnos que cursan esta área encontramos que no hay las suficientes enseñanzas de los medios técnicos para el buen desarrollo de las Aplicaciones Web ya que el adiestramiento se basa en lenguajes de programación sin manejar algunas normas de programación o estándares.

1.4 Objetivos

1.4.1 Objetivo general

Lograr difundir el desarrollo de Aplicaciones Web por medio del uso de Framework haciendo conciencia de los beneficios que trae trabajar con este.

1.4.2 Objetivos específicos

- Mencionar las ventajas de los Framework.
- Mostrar los diferentes Framework para distintos momentos del desarrollo de Aplicaciones Web.
- Usar Framework como base para desarrollo de Aplicaciones Web.
- Establecer parámetros de comparación que nos puedan permitir determinar con claridad que Framework se adapta mejor a un desarrollo de Aplicaciones Web.

1.5 Hipótesis

El hacer difundir y concientizar a los estudiantes de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Tuxtla Gutiérrez en el uso de un Framework para el desarrollo de Aplicaciones Web en la elaboración de su residencia lograremos que el alumno realice su trabajo con mayor facilidad y eficiencia por las ventajas que estos ofrecen, realizando un proyecto con posibles modificaciones para otras generaciones o dándoles continuidad por el mismo estudiante y no abandonarlo.

1.6 Justificación

Si difundimos el uso de un Framework en el desarrollo de Aplicaciones Web lograremos dejar en el pasado la forma desorganizada en que trabajan muchos de los estudiantes de Ingeniería en Sistemas Computacionales, al enseñarle un Framework para el desarrollo de Aplicaciones Web ayuda, el programador podrá tener un trabajo limpio, con el consecuente ahorro de tiempo, reutilización de código, seguridad en la aplicación, entre otras ventajas y sobre todo tener continuidad en el proyecto y no abandonarlo, o que otras generaciones mejoren o actualicen la aplicación.

1.7 Alcances y limitaciones

1.7.1 Alcances

- Si los estudiantes adoptan un marco de desarrollo podrán reducir el tiempo en que realizan la Aplicación Web.
- Al usar un Framework se trabaja con la arquitectura Modelo – Vista – Controlador presentando un trabajo ordenado.
- Se logra un ahorro de líneas de código por las bibliotecas que contienen los Frameworks.
- Seguridad en la Aplicación Web.
- Reutilización de código.

1.7.2 Limitaciones

- El aprendizaje de un Framework puede ser prolongada en tiempo en el caso que el estudiante no tenga conocimientos de esto.
- El uso de un Framework no asegura el desarrollo de la aplicación si el desarrollador carece de lógica de programación.

1.8 Estado del arte

1.8.1 Sistemas desarrollados en la Ingeniería de Sistemas Computacionales del ITTG.

A continuación, se presentan algunos casos de estudio desarrollados por alumnos egresados que presentaron sus residencias profesionales.

1.8.1.1 Proyectos con Framework

1.8.1.1.1 Sistema para la Administración y Publicación en Línea de Calificaciones de la Escuela de Terapia Física del DIF Chiapas.

Es un sistema realizado por alumnos de la carrera de la Ingeniería en Sistemas Computacionales del ITTG (interfaz del sistema se muestra en la siguiente Ilustración 1.) como parte de proyecto de residencia Enero – Julio de 2013.



Ilustración 1: Interfaz de Sistema para la Administración y Publicación en Línea de Calificaciones de la Escuela de Terapia Física del DIF Chiapas.

El tiempo estimado aproximadamente es de 6 meses del desarrollo. Esta aplicación Web fue desarrollada para programar en ZendFramework, utilizando los lineamientos que este maneja MVC (Modelo – Vista – Controlador) usando este estándar para tener una buena organización.

También utiliza Bootstrap que es un Framework de CSS para darle estilo a la Aplicación Web, también usa JQuery y el uso de un Gestor de Base de Datos (MySQL).

1.8.1.1.2 Sistema Web de Inventario de la Intranet de CFE de la Zona de Distribución de Tuxtla

Un sistema realizado por alumnos de la ingeniería de sistemas computacionales del ITTG (interfaz del sistema se muestra en la siguiente Ilustración 2), como parte de residencia de enero - julio 2014.

The image shows a web browser window displaying a login page. At the top, there is a green header bar with the text "Sistema Web de Inventario de la Intranet" and "CFE COMISIÓN FEDERAL DE ELECTRICIDAD". Below the header, the text "Departamento de Tecnologías de la Información" is visible. The main content area features a rounded rectangular box with the title "INICIAR SESIÓN". Inside this box, there are two input fields: "Username *" with the value "Usuario" and "Password *" with the value "password". Below the password field is a checkbox labeled "Recordarme" and a button labeled "Ingresar". At the bottom of the page, there is a small copyright notice: "Copyright © 2014 CFE. Todos los derechos reservados."

Ilustración 2: Interfaz de Sistema Web de Inventario de la Intranet de CFE de la Zona de Distribución de Tuxtla

El tiempo estimado aproximadamente es de 6 meses del desarrollo. Esta Aplicación Web es desarrollado en un marco de trabajo Yii Framework bajo el lineamiento Modelo-Vista-Controller, además de usar otras herramientas para su elaboración como CSS, JavaScript, JQuery, entre otros, esto para darle estilos, eventos, interacciones, etc., también un gestor de base de datos que es MySQL.

1.8.1.2 Proyectos sin Framework

1.8.1.2.1 Sistema de Administración de Servicios Catastrales.

Es un sistema realizado por alumnos de la carrera de la Ingeniería en Sistemas Computacionales en el Instituto Tecnológico de Tuxtla Gutiérrez (interfaz se muestra en la ilustración 3.), como parte de su residencia y titulación integrada en el periodo de agosto – diciembre de 2013.



Ilustración 3: Interfaz de Sistema de Administración de Servicios Catastrales.

El tiempo estimado es de aproximadamente 6 meses de desarrollo. Durante el trabajo de esta Aplicación Web fue seleccionado para programar en el lenguaje ASP, pero sin ningún lineamiento en la cual probablemente utilicen parte de la programación para realizar los procesos dentro del sistema, todo esto con el fin de usar el editor Microsoft Visual Studio, además de que esta tecnología está limitada ya que requiere de un servidor IIS y solo funciona con Microsoft Windows.

En cuanto a la organización, se ha encontrado una carpeta de funciones en donde se encuentran todas las rutinas utilizadas dentro del sistema, entre estas se encuentra la rutina de acceso al sistema, salida del sistema, probablemente se encuentre la configuración de acceso a la base de datos, entre otros. También cuenta con otras carpetas en donde se almacenan archivos CSS, Imágenes y rutinas JavaScript.

1.8.1.2.2 Desarrollo de un Sistema de Información en la Coordinación de Agenda Logística del Estado de Chiapas (Módulo Viáticos)

Desarrollo de un sistema realizado por alumnos de la carrera de Ingeniería en Sistemas Computacionales en el ITTG (interfaz se muestra en la siguiente imagen), como parte de la residencia profesional en el periodo de enero - julio de 2012.



Ilustración 4: Interfaz Desarrollo de un Sistema de Información en la Coordinación de Agenda Logística del Estado de Chiapas (Módulo Viáticos)

El tiempo estimado del desarrollo de este sistema fue de aproximadamente 6 meses. En el trabajo de este proyecto se utilizó una programación web de cero en el lenguaje PHP, esto hace que no conlleve a una organización adecuada además de que no hay un lineamiento en lo cual puede tener problemas en un futuro (actualizaciones), también usan JavaScript y como gestor de base de datos usan MySQL, pero como ya se había mencionado con anterioridad sin ningún lineamiento y un trabajo de programación completamente desordenado.

1.8.1.2.3 Sistema de Control de Inventarios de Reactivos que se Surtirán en el PTNDIPAB (Polo Tecnológico Nacional de Desarrollo de Investigación en Pruebas Analíticas y Biocombustible)

El sistema fue creado en un tiempo estimado de 6 meses en el cual se desarrolló con el fin de trabajar en la residencia profesional (como se muestra en la siguiente imagen). El proyecto se trabajó sin ningún reglamento ni lineamiento usando un lenguaje de programación web como PHP usando también HTML 5 y CCS para darle estilo, diseño, etiquetas, etc., usando así otro lenguaje de programación como JavaScript y Ajax para el manejo de formularios para facilitar el control y manejo de información entre la aplicación web y el servidor. Para el manejo de la base de datos utilizo MySQL para poder almacenar datos del proyecto.

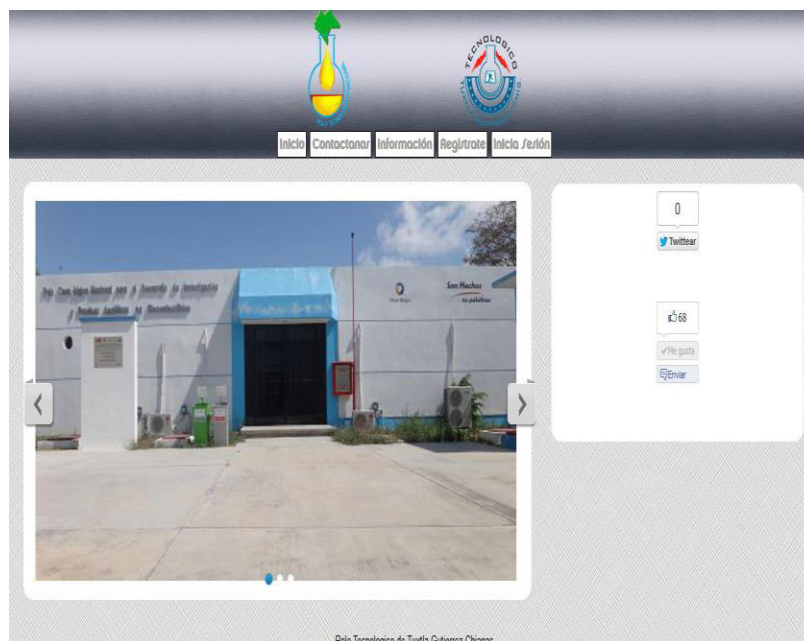


Ilustración 5: Interfaz de Sistema de Control de Inventarios de Reactivos que se Surtirán en el PTNDIPAB

Capítulo II: Marco teórico

2.1 Marco teórico técnico conceptual.

El marco teórico técnico está compuesto por las definiciones de palabras y/o frases clave que son mencionadas dentro del documento, además estas se encuentran organizadas en orden alfabético.

AJAX: Asynchronous Javascript and XML es una técnica de desarrollo Web que genera Aplicaciones Web interactivas combinando: XHTML y CSS para la presentación de información, Document Object Model para visualizar dinámicamente e interactuar con la información presentada, XML, XSLT para intercambiar y manipular datos, XMLHttpRequest para recuperar datos asíncronamente y Javascript como nexo de unión de todas estas tecnologías. (Eguíluz Pérez, 2008)

Bootstrap: Es el Framework de Twitter que permite crear interfaces Web con CSS y Javascript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automática se adapta al tamaño de un ordenador o de una Tablet sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo o Responsive Design. Ofreciendo todas las posibilidades a la hora de crear interfaces Web, los diseños creados con bootstrap son simples. Limpios e intuitivos, esto le da agilidad a la hora de cargar y al adaptarse a otros dispositivos. (Ramírez, 2014)

Blueprint: Es un marco CSS, que tiene como objetivo reducir el tiempo de desarrollo. Se le da una base sólida para construir su proyecto en la parte superior de, con una tabla fácil de usar, tipografía sensible, plugins útiles, e incluso una hoja de estilo para la impresión. (Blueprintcss, 2014)

CSS: Hojas de Estilo en Cascada (CSS) son un mecanismo sencillo para añadir estilo (por ejemplo, tipos de letra, colores, espaciado) a los documentos Web. (W3 org, 2014)

Codeigniter: Es un poderoso Framework PHP con un tamaño muy reducido, construido para programadores PHP que necesiten una herramienta simple y elegante para crear Aplicaciones Web con todas las funciones. (EllisLab, 2013)

Django: Es un Framework web Python de alto nivel que fomenta el rápido desarrollo y diseño limpio y pragmático. Desarrollado por una operación en línea-noticias de rápido movimiento, Django fue diseñado para manejar dos retos: los

plazos intensivos de una sala de redacción y los estrictos requisitos de los desarrolladores web con experiencia que lo escribieron. Permite construir de alto rendimiento, Aplicaciones Web elegantes rápidamente. (Django, 2014)

Framework: Se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Los objetivos principales que persigue un Framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un Framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo, clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. (Gutiérrez., 2014)

HMVC (Hierarchical-Model-View-Controller): Fue presentado en el año 2000 por Javaworld como una forma de dar solución a algunos de los problemas que se presentan al utilizar MVC en proyectos de desarrollo de gran tamaño.

Es una adaptación al MVC, el paradigma HMVC pretende solucionar algunos de los problemas que no soluciona el MVC (gestión de datos, gestión de eventos y flujo de aplicaciones). Mediante el trabajo de campo se ha desarrollado este patrón, el cual ofrece un potente y fácil entender de la metodología de diseño por capas para el desarrollo de una capa de presentación completa. Mientras que el MVC proporciona un marco eficiente para el desarrollo de la interacción GUI, HMVC escala al nivel de cliente completo. Algunos de los beneficios clave de la responsabilidad basada en la arquitectura de capas incluyen:

- Definición de comunicación entre capas y aislamiento de capas más altas.
- Define la comunicación entre capas con acoplamiento mínimo.
- La localización de la exposición al código de terceros.

Algunos de los beneficios clave del uso de HMVC revelan los beneficios de la orientación a objetos. Una arquitectura en capas de manera óptima:

- Reduce las dependencias entre partes dispares del programa.
- Aliente a la reutilización de código, componentes y módulos.
- Aumenta la capacidad de mantenimiento mientras que facilita la extensibilidad. (JavaWorld, 2000)

HTML: Es un lenguaje de marcado que le indica al navegador como debe presentarse el contenido. Los enlaces a otros documentos pueden estar presentes

en su contenido, proveyendo un método sencillo para navegar entre documentos en internet. (Vazquez Esteban, 2013)

HTTP: Siglas de Hypertext Transfer Protocol (Protocolo de transferencia de Hipertexto) es un protocolo de nivel de aplicación para sistemas distribuidos y colaboración. Es genérico, protocolo sin estado, que se puede utilizar para muchas tareas más allá de su uso para hipertexto, como servidores de nombres y sistemas de administración de objetos distribuidos. (W3 org, 2002)

Lighttpd: es un servidor web de alta velocidad y seguridad en la cual su arquitectura está orientada a eventos optimizados a un gran número de conexiones paralelas. (light, 2012)

Mongrel: Es un servidor Web pequeño, rápido y muy seguro, que se encarga de servir peticiones http, especialmente para aplicaciones Ruby, integrándose muy bien con el marco de trabajo Rails. Está desarrollado en su mayoría en lenguaje Ruby, pero utiliza algunas extensiones RubyC para el análisis sintáctico del URI (identificador uniforme de recurso) lo que lo hace más veloz. (Ospina Torres & Suárez Díaz, 2008)

MVC (Modelo-Vista-Controlador): Es uno de los patrones más citados. Comenzó como un Framework desarrollado por Trygve Reenskaug para la plataforma Smalltalk en 1970. Desde entonces, ha desempeñado un rol influyente en más Frameworks de User interface y en el pensamiento acerca del diseño.

Como trabaja:

El MVC considera tres papeles: el modelo, el controlador y la vista, como se muestra en la ilustración 6.

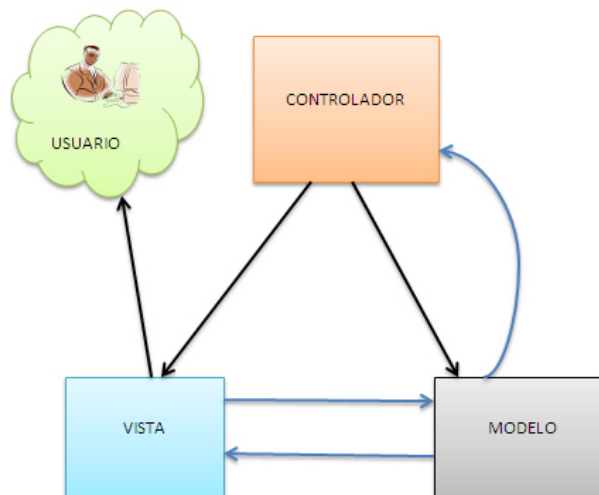


Ilustración 6: Representación del funcionamiento del MVC

- **Modelo:** Es un objeto que representa alguna información acerca del dominio. Este es un objeto no visual que contiene todos los datos y comportamientos distintos a los utilizados por la interfaz de usuario. En su forma más pura Orientado a Objetos el modelo es un objeto dentro del Modelo del Dominio.
- **Vista:** Define exactamente que es presentado al usuario. Normalmente los controladores pasan los datos para ser representados en la vista con algún formato. Las vistas también recopilan datos del usuario, con formularios.
- **Controlador:** Los controladores integran el patrón siempre. Ellos manipulan a los modelos, decide que va a mostrarse basado en la petición del usuario y en otros factores, pasa los datos que cada vista necesita, o sede el control completo a otro controlador. (Zend, 2014)

MySQL: Es un sistema de gestión de bases de datos SQL Open Source, lo desarrolla, distribuye y soporta MySQL AB. El servidor de Base de Datos de MySQL es rápido, fiable y fácil de usar, es capaz de gestionar Base de Datos Relacionales en donde los datos se encuentran distribuidos en múltiples tablas. Es Open Source, lo que significa que es posible para cualquier persona utilizar y modificar el software para adaptarlo a sus necesidades. (MySQL, 2011)

PHP: Acrónimo de "*PHP: HypertextPreprocessor*", es un lenguaje "*Open Source*" interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser incrustado en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP. (The PHP Documentation Group, 2012)

Phusion passenger: Es un servidor web y servidor de aplicaciones para sus Aplicaciones Web. (passenger, 2012)

Programación orientada a objeto: La programación orientada a objetos es una metodología que descansa en el concepto de objeto para imponer la estructura modular de los programas. Permite comprender el dominio del problema a resolver, al intentar construir un modelo del mundo real que envuelve nuestro sistema. Es decir, la representación de este mundo mediante la identificación de los objetos que constituyen el vocabulario del dominio del problema, su organización y la representación de sus responsabilidades. (Lazzarini, 2012)

Prototype: Es la librería fundamental de esta implementación, escrita en JavaScript. Incluye un conjunto de extensiones para este lenguaje, para el navegador web y para el objeto XMLHttpRequest. (Camargo Apen, 2008)

Ruby on Rails: Es un entorno de desarrollo Web de código abierto que esta optimizado para la satisfacción de los programadores y para la productividad sostenible. Te permite escribir un buen código evitando que te repitas y favoreciendo la convención antes que la configuración. Rails es un almacén para construir Aplicaciones Web que acceden a base de datos. Un conjunto de librerías, automatismos y convenciones destinados a resolver los problemas más comunes a la hora de desarrollar una Aplicación Web, para que el programador pueda concentrarse en los aspectos únicos y diferenciales de su proyecto en lugar de en los problemas recurrentes. (Rails, 2013)

WampServer: Es un entorno de desarrollo web para Windows. Permite crear Aplicaciones Web con Apache 2, PHP y una base de datos MySQL. Además, cuenta con phpMyAdmin que permite manejar fácilmente las bases de datos. (WampServer, 2014)

WEBrick: Es un conjunto de herramientas de servidor HTTP que puede ser configurado como un servidor HTTPS, un servidor proxy y un servidor virtual-host. Webrick cuenta con el registro completo de las dos operaciones de un servidor y acceso HTTP. (WEBrick, 2013)

XAMPP: Es el entorno más popular de desarrollo con PHP, es una distribución de Apache completamente gratuita y fácil de instalar que contiene DB, PHP y Perl. El paquete de instalación ha sido diseñado para ser increíblemente fácil de instalar y usar. (XAMPP, 2015)

Zend Framework: Es un Framework open source (código abierto) para el desarrollo de aplicaciones y servicios web con PHP 5. Zend Framework está implementado usando código 100% orientado a objetos. La estructura de componentes de Zend Framework es algo única; cada componente está diseñado con poca dependencia en otros componentes. Esta arquitectura de bajo acoplado permite a los desarrolladores usar los componentes individualmente, este diseño es llamado “utilizar a voluntad”. (Zend Framework, 2014)

Capítulo III. Análisis de Framework

3.1. Ruby On Rails



Ilustración 7: Ruby On Rails

Generalmente es un Framework en la cual nos permite crear Aplicaciones Web usando una arquitectura para interactuar con una aplicación de Rails, un Navegador Web envía una solicitud, la cual es recibida por un Servidor Web y se transmite a un controlador de Rails, que es el encargado de definir qué hacer a continuación. El controlador interactúa con un modelo, que es un objeto de Ruby que representa un elemento de la página (por ejemplo, un usuario) y se encarga de comunicarse con la base de datos. Después de invocar el modelo, el controlador, genera la vista (tal como se muestra en la Ilustración 9) y devuelve la página web completa en el navegador como HTML.

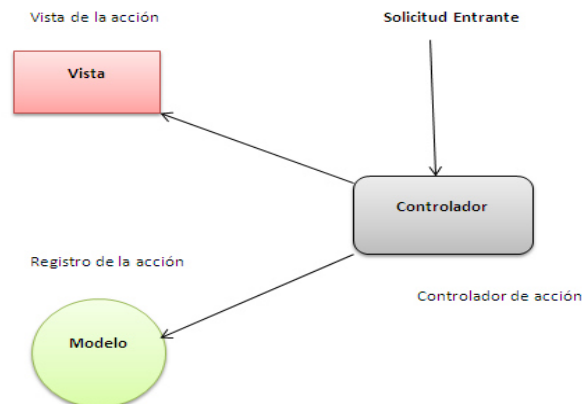


Ilustración 8: Componente de Rails y MVC

Rails proporciona diferentes facilidades que hacen más fácil implementar aplicaciones Ajax, por lo tanto, esto permite procesar peticiones de un Navegador Web a un Servidor Web que procesa en segundo plano sin cargar otras páginas Web adicionales.

Para desarrollo y pruebas, se utiliza Mongrel o WEBrick los Servidores Web con lo cual trabaja el Framework de Ruby on Rails. Para utilizar Rails en otros servidores en producción se está extendiendo el uso de Passenger. Otra opción para producción son Nginx, Mongrel, Apache o alguna combinación de ambos (por ejemplo, utilizando Apache como proxy para los procesos Mongrel). Aunque su uso no se recomienda porque no es seguro utilizar múltiples aplicaciones Ruby On Rails sobre Apache.

Dada que la arquitectura Rails se recomienda usar un Sistema de Gestión de Base de Datos Relacional (SGBDR) para almacenamiento de datos.

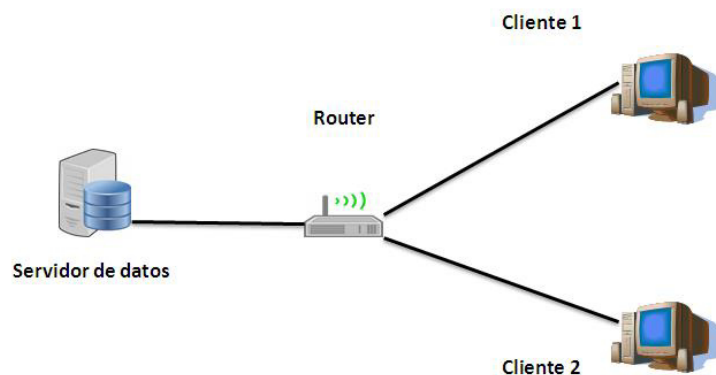


Ilustración 9: Servidor de datos Cliente / Servidor

Rails soporta la biblioteca SQLite por defecto. El acceso a la base de datos es totalmente abstracto desde el punto de vista del programador, es decir que es agnóstico a la base de datos, y Rails gestiona los accesos a la base de datos automáticamente. Rails puede soportar diferentes SGBDR, incluyendo MySQL, PostgreSQL, SQLite, IBM DclienteB2 y Oracle.

3.2. CodeIgniter



Ilustración 10: Logotipo de CodeIgniter

Es un Framework para crear Aplicaciones Web trabaja bajo la arquitectura Modelo – Vista – Controlador, aunque este marco utiliza una forma más peculiar al modo de operar de la arquitectura tal como se logra ver en la siguiente imagen.

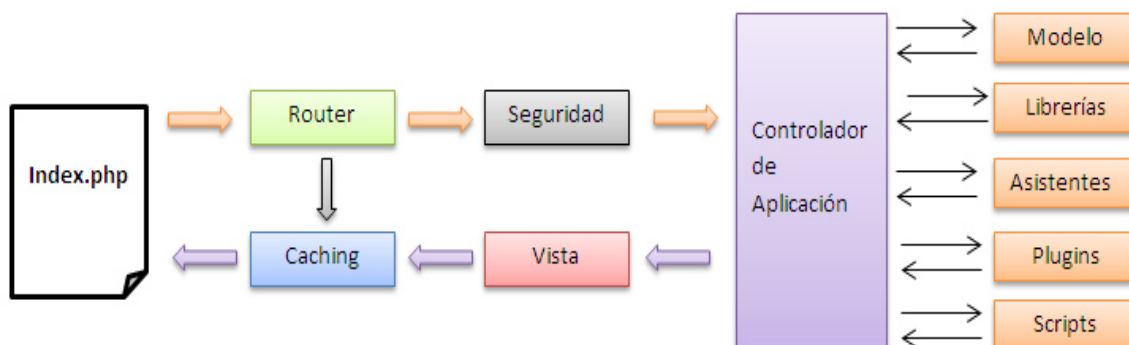


Ilustración 11: Diagrama de Flujo de CodeIgniter

1. El index.php sirve como controlador frontal, inicializando los recursos básicos necesarios para correr CodeIgniter.
2. El Router examina la petición HTTP para determinar que debe ser hecho con él.
3. Si un archivo de cache existe, es enviado directamente al explorador, sobrepasando el sistema de ejecución normal.
4. Seguridad. Antes que el controlador sea cargado, la petición HTTP y cualquier dato suministrado por el usuario es filtrado por seguridad.
5. El controlador carga los modelos, librerías, plugins, asistentes y cualquier otro recurso necesario para procesar la petición específica.

6. La vista finalizada es presentada entonces enviada al explorador web para ser observada por el usuario.

Para trabajar con un Servidor Web necesitamos disponer de un servidor PHP 4 ó PHP 5. Ambos funcionan perfectamente y en el caso de PHP 4, la versión mínima que debemos tener es la PHP 4.3.2.

Por lo que respecta a las bases de datos, CodeIgniter es compatible con las más habituales en el desarrollo de Aplicaciones Webs: MySQL (4.1 o posterior), MySQLi, MS SQL, Postgres, Oracle, SQLite.

En la siguiente imagen se puede observar la carpeta de sistema y como están dividido las carpetas en la cual se requiere trabajar para poder generar una Aplicación Web, por lo tanto, significa que es en donde estará el motor del Framework.

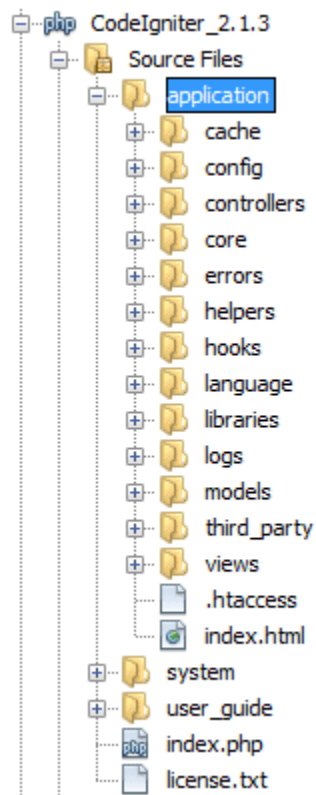


Ilustración 12: Estructura de carpetas del Framework CodeIgniter

En la ruta System/application es donde tenemos que crear los controladores, vista y todo el funcionamiento de la Aplicación Web. Todo lo que se modifique fuera de esta ruta, puede afectarnos cuando hagamos actualizaciones del Framework o de la Aplicación Web.

Dentro de esa carpeta, nos podemos encontrar lo siguiente:

- Config: Ficheros de configuración del propio codeigniter o de nuestras clases.
- Controller: Creamos nuestros controladores.
- Error: Clases para poder gestionar los errores.
- Helpers: Funciones que nos ayudan a mostrar o generar el contenido.
- Hooks: Son funciones que le podemos dar la orden que se carguen, por ejemplo, antes de cargar los controladores o que sea la última función en cargar.
- Language: Para las traducciones.
- Libraries: Donde podemos guardar nuestras propias clases para utilizarlas en los controladores.
- Models: Clases para trabajar con la capa de base de datos.
- Views: Donde se crean la vista para mostrar el contenido.

3.3. Django



Ilustración 13: Logotipo de Framework django

Una aplicación basada en la tecnología Web y desarrollada en Python, y esto implica de manera directa la elección de Django. Django es un Framework de Aplicación Web de código abierto, que nos permite construir Aplicaciones Web más rápidamente, con un diseño limpio y pragmático.

Python es usado en todas las partes del Framework, incluso configuraciones, archivos y en los modelos de datos. Django hace hincapié en el re-uso, el desarrollo rápido, la conectividad y extensibilidad de componentes, y en el principio “No te repitas” (DRY, del inglés Don’t Repeat Yourself).

La arquitectura de Django es una variación del patrón Modelo – Vista – Controlador, el cual es una guía de alto nivel para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. La variación que realiza este Framework es Modelo – Plantilla – Controlador que intenta ser lo más funcional posible. El Modelo en Django sigue siendo Modelo; a la Vista en Django se le denomina Plantilla y en el caso del Controlador se le denomina Vista. Cómo se logra ver en la Ilustración 14.

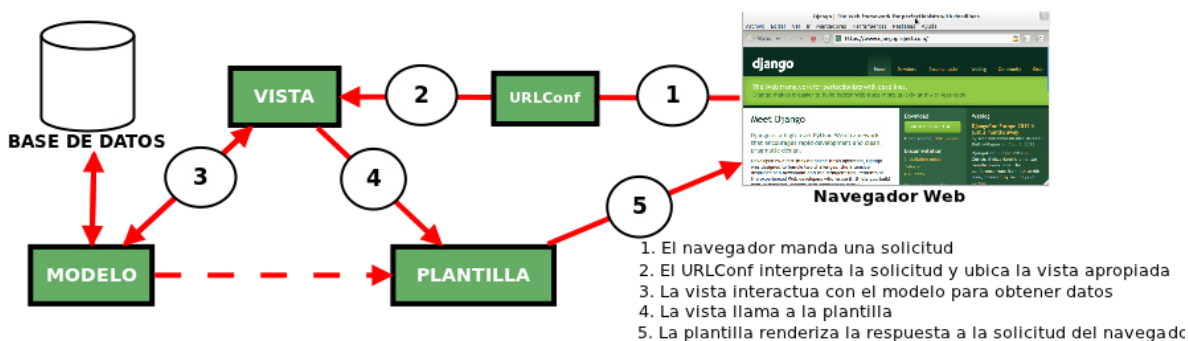


Ilustración 14: Funcionamiento de MPC Django

Modelo: El modelo define los datos almacenados, se encuentra en forma de clase de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros; también posee métodos. Todo esto nos permite indicar y controlar el comportamiento de los datos.

Controlador: La vista se presenta en modo de funciones en Python y su propósito es determinar qué datos serán visualizados. El ORM de Django permite escribir código Python en lugar de SQL para realizar las consultas que necesita la vista. La autenticación de servicios y la validación de datos a través de formularios son también tareas de las que se encarga la vista.

Plantilla: La plantilla es básicamente una página HTML con algunas etiquetas extras propias de Django, en sí no solo crea contenido en HTML sino también XML, CSS, JavaScript... La plantilla recibe los datos de la vista y luego los organiza para la presentación al navegador web.

Respecto a la base de datos, la recomendada es PostgreSQL, pero también son soportadas MySQL y SQLite 3. En el modelo de datos de Django, una clase representa un registro de una tabla en la base de datos y las instancias de esta serán las filas en la tabla.

Como mencionamos en los requisitos, Django incluye un servidor web liviano para realizar pruebas y trabajar en la etapa de desarrollo. En la etapa de producción, sin embargo, se recomienda Apache 2 con mod_python. También puede soportar otros Servidores Web como FastCGI o SCGI en Apache y Lighttpd.

Django se puede resumir que tiene muchas facilidades en las cuales las ventajas principales son:

- Es rápido de desarrollar.
- Es modular.
- Tiene muy bajo acoplamiento.
- Genera automáticamente un panel de administrador.
- Sus bibliotecas hacen gran parte del trabajo.
- Soporta varias bases de datos.
- Es Modelo – Plantilla – Controlador que son los patrones que Python desarrollo para este Framework. Aunque no cambia mucho al tradicional Modelo – Vista – Controlador.

3.4. Zend Framework



Ilustración 15: Logotipo de Framework Zend Framework

Zend Framework es un marco para el desarrollo de Aplicaciones Web utilizando el lenguaje PHP 5 es totalmente Open Source (código abierto), contiene librerías que ayudan al manejo de Base de Datos (DAO – Data Access Object), Sesiones, Autenticación, Validación, Filtros, Manejo de correos, Internacionalización, Webservice (XML RPC o REST) entre otros. La filosofía utilizada en Zend es escribe una vez y utiliza muchas veces, por lo que se dedica a que el desarrollador se concentre en reutilizar código.

Para Zend Framework existen dos formas de utilizar, la primera consiste en usarlo como una librería que provee los componentes para desarrollar aplicaciones, la segunda consiste en utilizarlo como un marco de trabajo para el desarrollo de Aplicaciones Web con el propósito de sacarle provecho.

Por lo consiguiente esto tiene su marco de trabajo en la cual está constituida por una estructura especial de directorios y archivos, que permiten generar aplicaciones Modulares y aplicando Modelos – Vista – Controlador.

En la siguiente imagen podemos observar la estructura de las carpetas con la que trabaja este Framework la cual nos permite un mejor desarrollo y entendible ya que se puede ver las carpetas principales en las cuales se trabaja controllers, models y views. El resto de las carpetas contiene el motor del Framework ya que contiene las ayudas, librerías, etc.

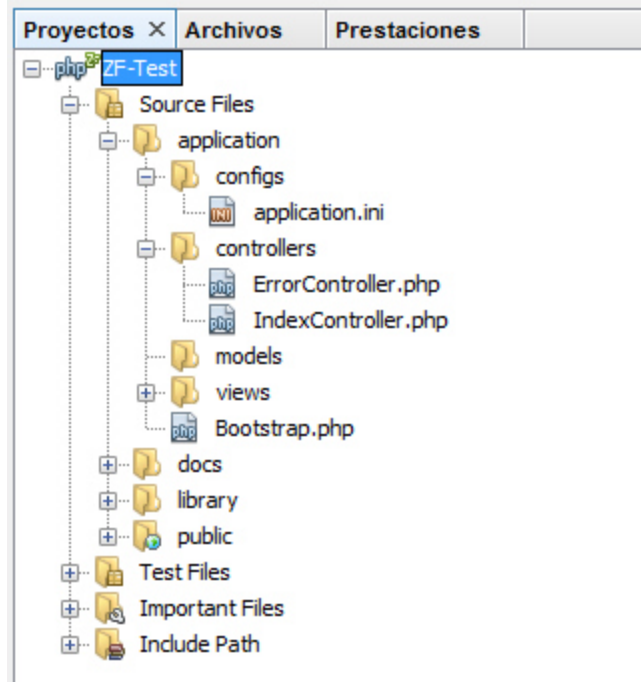


Ilustración 16: Estructura básica de una aplicación con Zend Framework

Pero si necesitamos hacer una aplicación más potente, con capacidad de extensión, de trabajo a largo plazo hay que optar por la estructura modular de Zend Framework, la cual está dada por la siguiente estructura de directorios.

La estructura básica modular permite construir módulos que se pueden agregar con el paso del tiempo, también contribuye a separar mucho más cada segmento del sistema por lo que el trabajo se torna mucho más organizado y el desarrollo puede ser incremental.

Cada módulo es independiente del otro por lo que si un módulo falla los demás no se verán afectados.

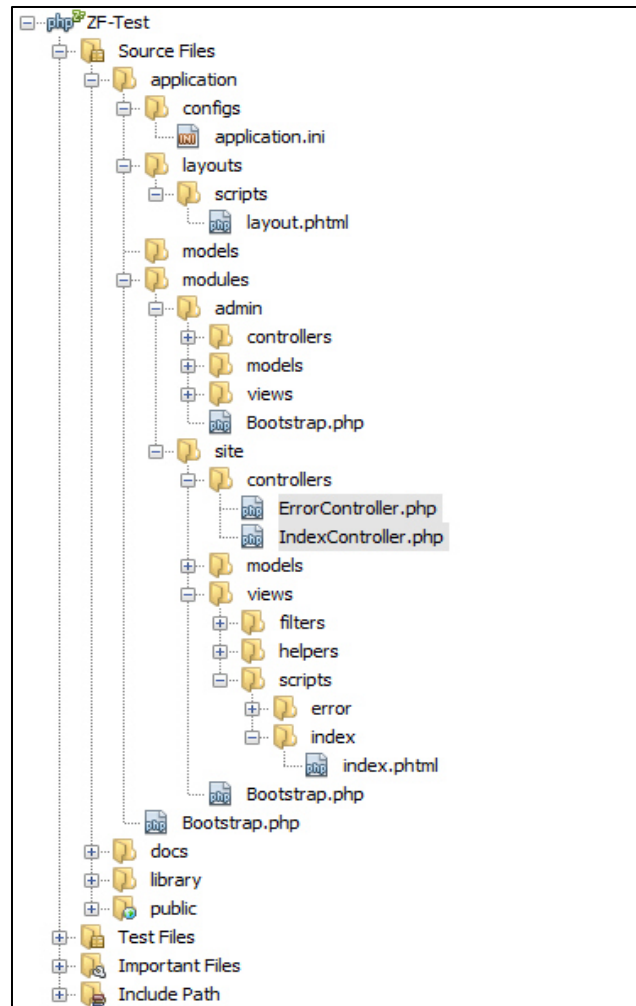


Ilustración 17: Estructura modular básica para ZF

Los componentes con los que cuenta este Framework pueden ser agrupados en las siguientes categorías:

- Infraestructura del núcleo del Framework: componentes requeridos por otros bloques del Framework, como memoria caché, configuración del ambiente de trabajo, operación por línea de comandos, registro de actividades y gestión de memoria, entre otros.
- Autenticación y autorización de acceso: responsables de la configuración de listas de control de acceso, autenticación de usuarios y manejo de sesiones.
- Base de datos: clases de acceso, modificación de tablas, obtención de datos mediante consultas SQL y lectura en formato XML.

- Internacionalización y Localización: para configurar la fecha y hora, la ubicación geográfica que tomará como base la aplicación web, las unidades de medida a usar y la posibilidad de traducir la información a otros idiomas.
- Emails, formatos y búsquedas: generación de archivos PDF, mensajes de correo electrónico en formato de texto y MIME (Multi-Purpose Internet Mail Extensions, Extensiones de correo Internet multipropósito).
- Web y Servicios Web: permite que la Aplicación Web pueda hacer uso de Servicios Web ofrecidos por aplicaciones externas y exponga servicios web propios.

Zend Framework referencia a lo fácil que es manejar la arquitectura modelo – vista - controlador y no solo eso a continuación se mostrara algunas de las ventajas:

1. Se pueden ampliar las clases.

Zend Framework está totalmente orientado a objetos, y por ello usa muchos conceptos de la orientación a objetos como son herencia e interfaces. Esto permite al desarrollador implementar sus propias variantes en los componentes sin tener que modificar el código base.

2. La programación de orientación a objetos.

En Zend Framework todo es un objeto. La principal ventaja de esto es hacer al código reusable.

3. Usar lo que necesitas.

Por diseño Zend Framework es una simple agrupación de clases. Usualmente se usa todo el potencial de Zend Framework para la construcción de proyectos plenamente funcionales, pero hay ocasiones en que solo se necesita cierto componente. Zend Framework permite el uso de cualquiera de sus librerías en vez de usarlo totalmente.

4. Tiene componentes muy importantes.

Una de las cosas más importantes de Zend Framework es que contiene muchísimos componentes. Por ejemplo. Zend_Auth: Aumentar un usuario, Zend_Acl: Controlar acceso a los recursos, Zend_Form: crear formularios. Zend Framework permite usar cualquiera de sus componentes para realizar diferentes tipos de funcionalidades que se puedan necesitar en una aplicación.

5. **Directrices y normas.**

Zend Framework es desarrollado conjuntamente con una extensiva contribución de usuarios. Esto significa que:

- Cualquier contribución debe ser de tipo Contribute License Agreement.
- El código debe ser probado por una unidad de prueba como PHPUnit.
- El código debe estar apegado a los estándares de codificación.

6. **Facilidades de realizar pruebas (test-driven).**

El desarrollo con Test-driven es una técnica de programación que requiere al programador realizar pruebas para cada función que se codifique antes de codificar la misma función.

Escribiendo primero las pruebas el programador garantiza:

- Pensar en los posibles casos de uso para el código.
- Crear una lista de entradas y salidas.
- Hace más sencilla la refactorización de código.
- Más fácil pasar código de una persona a otra.

7. **Comunidad y documentación.**

Como varios Framework tienen ayudas para el entendimiento de cómo desarrollar Aplicación Web con un marco de trabajo en la actualidad existe grandes cantidades de documentación, tutoriales en su sitio oficial, e inclusive videos tutoriales para ser mejor el aprendizaje de este Framework.

3.5. Bootstrap: Framework para CSS



Ilustración 18: Logotipo de Bootstrap

Es un Framework que sirve para realizar el diseño de una Aplicación Web, ya que este le permite al programador dar imagen y vista a una página web, teniendo muchas opciones para poder diseñar según el desarrollador lo requiera.

El Framework está indicado para desarrolladores y diseñadores novatos, si necesita profundizar en el código, se necesita el conocimiento de HTML, CSS y JavaScript/JQuery. Está construido para soportar varios navegadores y múltiples dispositivos.

Los Diseño de páginas web para que el usuario las visualice perfectamente en un amplio rango de dispositivos como: Navegador en el PC, tableta, Smartphone.

Bootstrap 3: Mobile first Se ha rediseñado para adaptarse desde el principio a tamaños de pantalla típicos de los móviles con un sistema grid fluido que escala a 12 columnas al aumentar el tamaño del viewport.



Ilustración 19: Tipos de dispositivos al que se adecuan Bootstrap

La distribución básica en la cual se trabaja en Bootstrap se observa en la siguiente imagen.

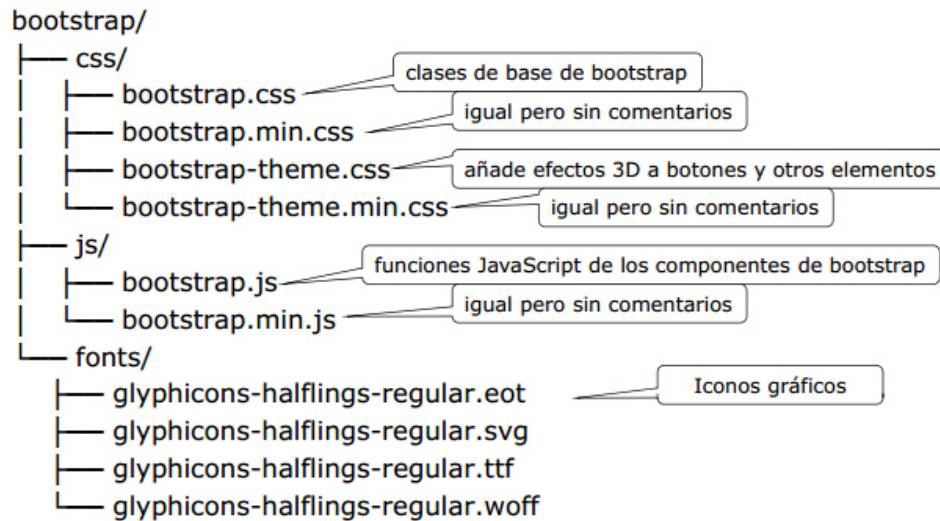


Ilustración 20: Distribución de los archivos de Bootstrap

La estructura que maneja Bootstrap como se observa en la siguiente ilustración

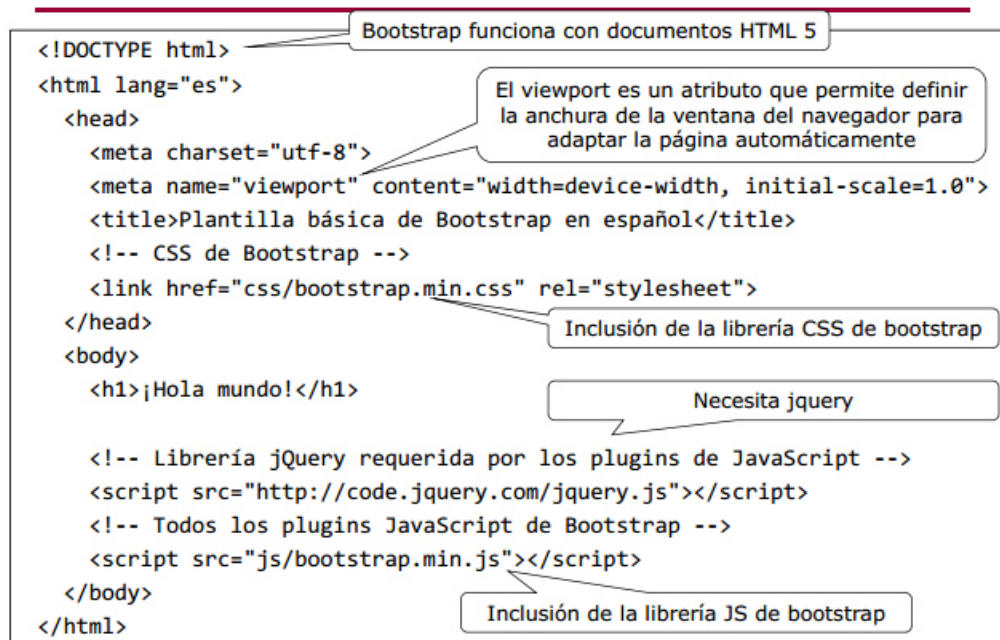


Ilustración 21: Estructura básica de Bootstrap

3.6 Blueprint: Framework de CSS



Ilustración 22: Logotipo de Blueprint

Es un Framework en la cual se trabaja el diseño de una Aplicación Web. También es una colección de archivos .css, los cuales poseen variados estilos predefinidos para ahorrar tiempo en diferentes tareas del diseño web: Maquetaciones, reseteo de estilos, diseño de formularios, tipografías, etc. Para la creación de la maquetación de la plantilla se utilizará su sistema de grillas.

El sistema de grillas de BluePrint es muy fácil de utilizar, permitiendo agilizar de forma severa el proceso de maquetación. Su mecanismo permite crear un máximo de 24 columnas (o grillas) las cuales se pueden combinar dependiendo de las necesidades.

las principales características con las que podemos trabajar con este Framework son:

Una rejilla, que nos permitirá crear cualquier estructura de página web. Básicamente se trata de un conjunto de clases CSS para posicionar cualquier elemento en un espacio de 950 píxeles de anchura, dividido en 24 columnas.

Una definición de tipografía predeterminada, de manera precisa para todos los elementos de la página que pueden tener texto.

Estilos para formularios, con los que mejorar las interfaces de usuario.

Estilos para impresión, con las definiciones CSS más útiles para que las páginas se lean bien cuando se imprimen en papel.

Estilos específicos para IE, con los que resolver algunos de los problemas más comunes de Internet Explorer.

Estos son los principales archivos con los que se trabaja este Framework, en lo cual nos representa una mejor organización de trabajo.

screen.css: Posee estilos de reseteo, tipográficos, para formularios y para crear maquetaciones a través de grillas. Más adelante se explicará el funcionamiento de las grillas utilizando Blueprint.

print.css: Estilos que mejoran el formateo del sitio al momento de imprimir.

ie.css: Estilos específicos para Internet Explorer. La etiqueta [if lt IE 8] significa que el estilo se cargará únicamente para versiones anteriores a Internet Explorer 8.

/plugins/fancy-type/screen.css: Incorpora algunos estilos extras para la manipulación de tipografías en textos.

/plugins/joomla-nav/screen.css: Estilos específicos para el menú de navegación horizontal del sitio.

3.7. ASP.NET



ASP.NET

Ilustración 23: Logotipo de ASP.NET

Herramienta de desarrollo web comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, Aplicaciones Web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP).

En la actualidad una aplicación ASP.NET puede ejecutarse de dos formas distintas:

Aplicaciones cliente/servidor: Estas aplicaciones están típicamente en formato de ejecutables compilados. Estos pueden integrar una interfaz de usuario, Pero acostumbran ser menos gestionables y escalables que las demás aplicaciones.

Aplicaciones que utilizan el navegador: Dichas aplicaciones están caracterizadas por contar con una interfaz de web. La interfaz gráfica integra varias tecnologías, las cuales son el HTML, XHTML, scripting, etc., siempre y cuando el navegador que se esté utilizando soporte estas tecnologías.

ASP.NET por lo general trabaja con la arquitectura Modelo, Vista, Controlador (MVC) tal como se observa en la siguiente Ilustración, que como ya se ha mencionado anteriormente es para tener una organización en el sistema de aplicación web que se esté trabajando.

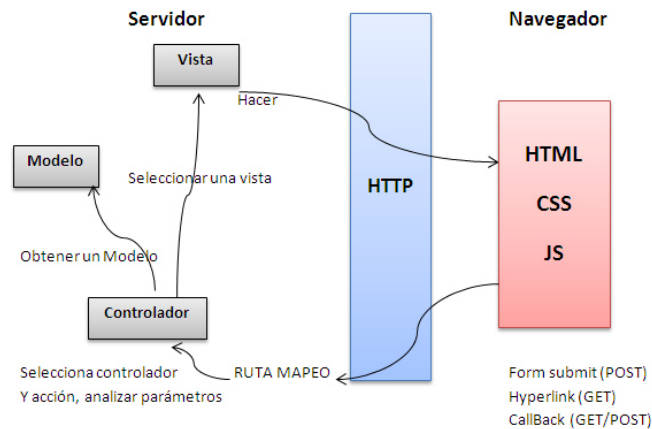


Ilustración 24: Arquitectura MVC de ASP.NET

Las características principales con las que podemos trabajar con ASP.NET ya que con esto nos favorece el desarrollo con este Framework.

- Pueden ser programados utilizando cualquier lenguaje del Framework .NET, como Visual Basic, C#, Managed extensions for C++, JScript.NET.
- Están contruidos sobre el Framework .NET y proveen de todos los beneficios de estas tecnologías, incluida la ejecución de un entorno controlado, de tipos de seguros y con herencia.
- Son soportados por Visual Studio, que ofrece herramientas RAD (Rapid Development) para (WebForms).
- Soportan un amplio de conjunto de controles basados en el servidor que ofrecen la posibilidad de desarrollar formularios estándar en un entorno de diseño RAD.
- Puede ser extendidos mediante controles creados por el desarrollador y por otras empresas.
- Las aplicaciones son independientes del navegador: WebForms ofrece un Framework o marco para crear la lógica de las aplicaciones en el servidor devolviendo al cliente texto HTML o XML, lo cual elimina la necesidad de codificar Microsoft, como el Internet Explorer 5 pueden aprovechar las características de un cliente web rico.

- Ofrece un modelo de programación basado en eventos: Esto permite implementar el mecanismo de eventos en las Aplicaciones Web de un modo transparente. El Framework .NET abstrae el modelo de eventos de modo que el mecanismo captura de un evento en el cliente, la transmisión al servidor y la llamada al método adecuado es automática y transparente.
- El modelo de objeto es abstracto e intuitivo: El Framework .NET para WebForms ofrece un modelo de objetos que permiten pensar en los Forms como en una unidad, en lugar de dos partes separadas de cliente y servidor. Los controles de WebForms de un servidor son una abstracción del contenido físico de una página HTML y de una interacción directa entre el navegador y el servidor del mismo modo que se utilizarían si fuesen de cliente y no es necesario preocuparse de crear el código HTML para presentar y procesar los controles y su contenido.
- Gestión de estado: el Framework ASP.NET gestiona de modo automático el almacenamiento de la información de estado del Form y de sus controles, y todo ello sin sobre utilizar los recursos.
- Funcionamiento de servidor escalable: el Framework ASP.NET permite escalar una aplicación desde un ordenador con un solo procesador hasta convertirla en una aplicación que funcione sobre WebForm con múltiples ordenadores sin realizar cambios complicados a la lógica de la aplicación.

Capítulo IV Análisis al desarrollar una Aplicación Web

Desarrollar Aplicaciones Web conlleva a realizar una serie de pasos antes de comenzar a escribir el código fuente de esta.

4.1 Aplicaciones web sin usar un Framework

Por lo general los estudiantes comienzan a aprender a programar Aplicaciones Web de libros o las enseñanzas de lo básico que adquieren de los maestros, inclusive de manuales tutoriales que se encuentran por la red, por lo que los alumnos solo buscan aprender lo básico y no hay un alineamiento o metodologías para el desarrollo de las paginas, conformándose con la información básica.

Un desarrollador novato normalmente para comenzar a desarrollar realiza las siguientes tareas:

- Instalar un servidor web local (WampServer).
- Utilizar un editor para poder trabajar (Dreamweaver).

También encontramos que desarrollan siguiendo una estructura similar a la siguiente:

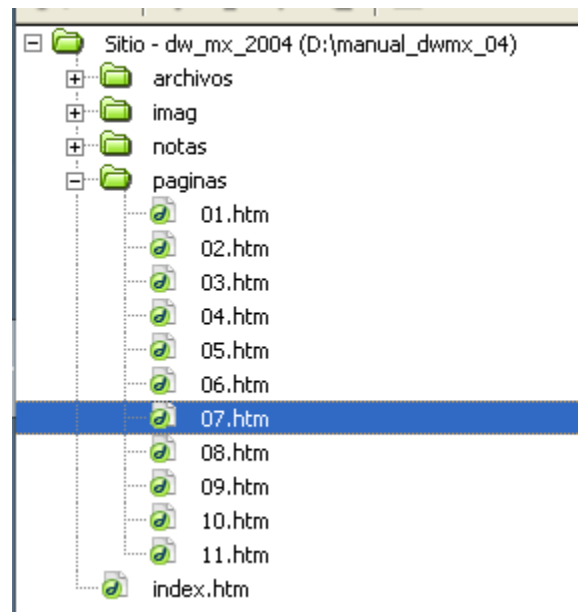


Ilustración 25: Organización de los archivos de las Aplicaciones Web

Los principales errores que se presentan en las aplicaciones que desarrollan los alumnos, es que haya incompatibilidad o diferencias donde se coloca la aplicación web es entre el servidor local y el servidor real.

Como ya se ha mencionado antes el tiempo de desarrollo de las Aplicaciones Web si un Framework es más tardío ya que estos tipos de trabajo se comienzan de cero realizando una programación artesanal.

4.2 Aplicaciones web usando Framework

En contraste podemos encontrar a los desarrolladores que comienzan a aventurarse con algún Framework como apoyo para el desarrollo de su aplicación, quienes realizan siguientes tareas antes de comenzar a desarrollar:

- Seleccionar el Framework y descargarlo.
- Instalar un servidor web local.
- Instalar un editor (Sublime).

Existen Framework para desarrollar en PHP y cada uno tiene su forma peculiar para la organización de sus archivos en la cual ya se mostró su estructura en el capítulo III de cada Framework, pero comparten la utilización de Modelo – Vista - Controlador.

Dependiendo del Framework que se haya escogido el método de instalación y utilización cambia.

Con el uso de un Framework para el desarrollo de una Aplicación Web nos permite reducir el tiempo, tener ayuda en el código fuente ya que podemos re-usar códigos mediante orientada a objetos y también permite el desarrollo de la aplicación y sus partes con mayor orden, en la cual este marco de trabajo utiliza una metodología le podríamos llamar así ya que trabaja por medio de Modelo – Vista - Controlador y permite que el código sea más entendible.

A continuación, una tabla comparativa de todos los Framework para el desarrollo de Aplicaciones Web.

	Ruby on Rails	CodeIgniter	Django	Zend	ASP.NET
Lenguaje de programación	Ruby	PHP	Python	PHP	Visual Basic
Tipo modular	x	x	✓	✓	x
Usa Modelo – Vista - Controlador	✓	✓	✓	✓	✓
Seguridad	✓	✓	✓	✓	✓
Estructura de carpetas	✓	✓	✓	✓	✓
Manejo de varios Bases de datos	✓	✓	✓	✓	✓

Bootstrap y Blueprint son Framework para el diseño de Aplicaciones Web en el cual su lenguaje de programación es CSS para darle estilo, presentación a una Pagina Web según el programador lo requiera esta también tiene una estructura en la cual permite el desarrollo de manera ordenada, tiene librerías para usar opciones de diseño y efectos según requiera la Aplicación Web.

Capítulo V Propuesta

Para resolver el problema de la falta de organización dentro de un proyecto y ayudar a su rápido desarrollo, es recomendable utilizar un Framework y seguir su estructura recomendada para la organización de los archivos.

En la actualidad hay muchos Framework, los cuales han sido mencionados son los más usados, ya que cada uno propone su forma de organizar las carpetas o archivos para el desarrollo de las Aplicaciones Web. Cada uno de ellos tiene su estructura en la cual se mencionó uno en específico que además cuenta con una estructura modular ya que en estas se pueden dividir una página en módulos y así hacer proyecto Web más ordenado y más fácil para el programador.

Una aplicación web se puede desarrollar en diferentes sistemas operativos como Windows, Linux o Mac OS, en lo cual se debe configurar un servidor Web en el cual poder ejecutar el código lo según lo requiera el Framework usado, para esto podemos usar paquetes pre-configuradas como: WampServer para Windows, xampp que se puede usar en los sistemas ya mencionados, lamp para Linux, MAMP para Mac OS. Estos son los más conocidos para poder trabajar en los sistemas operativos ya mencionados.

Realizar conferencias sobre el uso de Framework, en la actualidad está la materia de aplicaciones de internet en la cual se maneja el Framework de lenguaje ASP, en el cual será mejorar esos aprendizajes ya que se pueden aprender de otros Frameworks en la cual se puede trabajar.

También podrían haber diplomados que involucre al alumno del manejo de Framework u otros métodos en el cual el alumno se pueda involucran en el ámbito del desarrollo de Aplicaciones Web.

Conclusiones.

Analizando cada aplicación presentada se llega a los resultados, en estos se ven reflejados la utilización de cada estilo de programación, usar patrones de diseño o no, usar un Framework o no usarlo.

En la siguiente tabla se comparan algunos aspectos importantes al desarrollar en PHP y desarrollar usando un marco de Framework.

	php	Framework
Facilidad de aprendizaje	Menor tiempo	Mayor tiempo
Tiempo de desarrollo	Mayor tiempo	Menor tiempo
Separación de código HTML y PHP (MVC)	x	✓
Facilidad de cambiar entre sistemas de bases de datos (MySQL, Postgre, SQL Server)	x	✓
Sentencias SQL en cada página usada	✓	x
Programación orientada a Objetos	✓	✓
Facilidad de hacer cambios en sentencias SQL	✓	✓
Facilidad de hacer cambios en sentencias SQL de múltiples páginas	x	✓
Uso de patrones de diseño	x	✓
Reutilización de código	x	✓

También podemos comparar la forma en que se construye cada página según el estilo de programación.

La organización típica de proyectos de Aplicaciones Web sin usar un Framework intercalando código HTML y PHP.

Encabezado HTML: marca la estructura del código de HTML refiriéndose a las etiquetas que requiere.

Conexión a la base de datos (PHP): Aquí ya sea en el mismo archivo u otro archivo se coloca la programación para la conexión de la aplicación a la base de datos y pueda comunicarse entre si esto se coloca en lenguaje PHP.

Inicio de la página HTML: se requiere el cuerpo de la página diseño de la página y dándole un formato o un estilo propio en un lenguaje HTML

Consultas SQL a la base de datos: las consultas que se harán a la base de datos

HTML y tablas PHP: la combinación de lenguaje HTML y PHP sería la lógica de la aplicación

Pie HTML: cierre de la aplicación en HTML.

Por otro lado, encontramos el diagrama de flujo de cómo se organiza una aplicación creada con Framework.

Organización de una aplicación usando MVC con Framework

Archivo Index: a que se hace la conexión de la aplicación web con la base de datos.

Archivo controlador: aquí se programa toda la lógica de la aplicación web.

Archivo modelo: aquí se hacen las consultas que se necesitan para la interacción con la base de datos.

Archivo vista: aquí se hace las plantillas y el código aquí usamos HTML.

La ventaja de una arquitectura de Framework es que se logre una mejor organización del proyecto donde pueda haber modificaciones por otros desarrolladores independientes del creador de ese trabajo basándose en un marco de trabajo, con la facilidad de crear una Aplicación Web con el sistema modular como tal es el caso de Zend Framework permite que la aplicación tenga un crecimiento progresivo debido a lo ya mencionado a la organización.

Ya que el uso de un Framework está pensado para el desarrollo rápido de aplicaciones (Rapid ApplicationDevelopment o RAD) se logra una mejora en el tiempo de desarrollo de Aplicaciones Web a diferencia de escribir aplicaciones con PHP puro. Ya que con un marco específico es mucho más rápido, fácil, seguro y con el re-uso de código hace más sencillo la programación.

El uso de Frameworks es cada vez más importante para el mundo laboral y para el aprendizaje de los alumnos en base a encuesta como se muestra en el Apéndice A, que se realizó a los estudiantes de 8° semestre del Instituto Tecnológico de Tuxtla Gutiérrez, los resultados fueron tienen conocimiento de un Frameworks y algunos más de uno, pero obteniendo el aprendizaje de libros, tutoriales, videos, etc. en lo cual es magnífico, pero se puede mejorar el aprendizaje de estos. Más

sin embargo estos jóvenes si tienen interés de aumentar los conocimientos en caso de ser necesario.

En varias entrevistas que se les hizo a programadores de Aplicaciones Web se observa en el Apéndice A se comprueba que es mejor trabajar con los Frameworks para Aplicaciones Web ya que con estos presentan ventajas como reutilización de códigos, apoyo de librerías, ahorro de tiempo en programación, organización en la estructura, etc. Por lo tanto, el desarrollo en Aplicaciones Web con Framework obtenemos muchas facilidades.

Apéndice A

Entrevistas y resultados realizados a programadores web.

Se realizó una entrevista a 5 programadores dando sus opiniones y experiencias de los Framework para el desarrollo de Aplicaciones Web.

- **¿Recomendaría usar Framework para las Aplicaciones Web?**

El 100% recomendó usar

- **¿Por qué es necesario un Framework para Aplicaciones Web?**

El 60% de los programadores opino que tienen una estructura y organización a través de Modelo – vista – controlador.

40% la reutilización de código y ahorro de tiempo en el desarrollo.

80% otras opciones como los beneficios de las características de los Framework

- **¿Qué ventajas tendría al usar un Framework para Aplicaciones Web?**

El 100% respondió que la programación es estructurada y entendible con el Modelo – Vista – Controlador, reutilización de código y ahorro de tiempo.

60% menciona la seguridad.

60% dar mantenimiento (mejoras o modificaciones).

40% apoyo por parte de las librerías.

- **¿Las librerías que contienen los Framework ayudan en la programación? ¿porque?**

100% de las respuestas fueron similares con opiniones de que hay mucho código para usar, ahorra tiempo, etc.

- **¿Estaría poniendo en riesgo si hacen los desarrollos basándose de un Framework?**

80% respondió que los Framework son seguros debido a las características como las validaciones o la seguridad precargadas contienen.

Solo el 20% respondió que depende del Framework ya que no todos pueden ofrecer la misma seguridad.

- **¿Qué beneficios encontraríamos en los Framework al elaborar un sistema web con estos?**

El 100% de las Respuestas fueron mayor ahorro de tiempo y reutilización de código

- **¿Cuáles son las facilidades de trabajar con un Framework para el desarrollo web?**

Las facilidades de trabajar con un Framework, es que nos ahorra código, ya que cada Framework tiene su propio motor que hace todo lo demás.

100% ahorro de código, uso del modelo – vista – controlador.

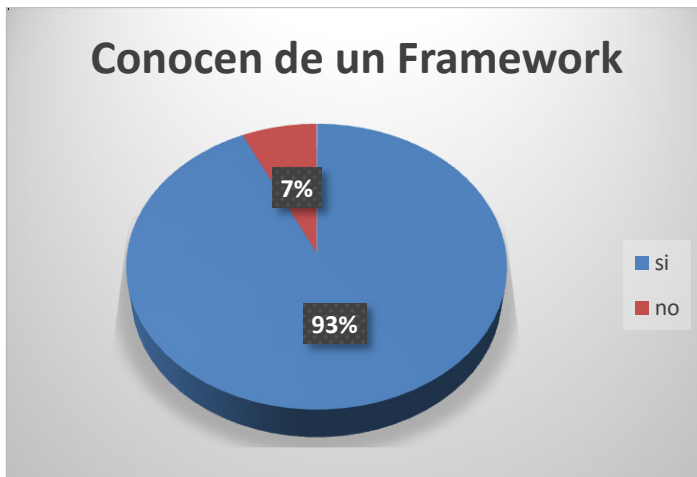
- **¿Qué diferencias podríamos encontrar en una programación sin Framework a una programación con Framework?**

100% concuerdan que la estructura que maneja un Framework lo hace más entendible y si no maneja Framework hace una estructura desordenada.

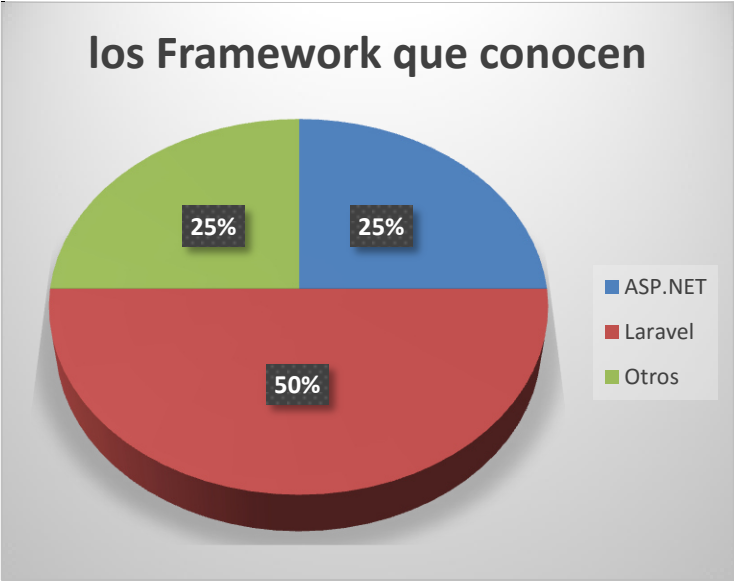
Preguntas y resultados de los Estudiantes de 8° semestre de tecnológico de Tuxtla Gutiérrez

Se realizó a un numero de 16 alumnos de 8° semestre en la cual los resultados son los siguientes en base a conocimientos, opiniones y experiencias

Estudiantes que si conocen de un Framework para desarrollar Aplicaciones Web
El 93% de los alumnos ha escuchado o conoce de un Framework
7% no conocen de un Framework



Framework que conocen
25% ASP.NET
50% laravel
25% yii, java script, Bootstrap, etc



Trabajos escolares trabajando en
 25% ASP.NET
 62% PHP
 13% C++, c#, CSS



Ayudas para obtener conocimiento
 50% tutoriales
 50% tesis, asesor académico, libros

Desean crecer conocimiento en Framework
 100% si

Ejemplo de las preguntas dirigida a los alumnos

- ¿Conoces o has escuchado de algún Framework para el desarrollo de aplicaciones web?
SI NO
- En caso de si ¿Cuál o cuáles Framework para desarrollo web conoces?
ASP, Java Script
- Actualmente tus desarrollos de proyectos escolares QUE IMPLIQUEN DESARROLLO de aplicaciones web ¿en qué lenguaje has trabajado?
Visual C#, Java y C++
- ¿Qué tipo de ayuda utilizas en el desarrollo de aplicaciones web?
- ¿Crees que sería de ayuda trabajar con un Framework de aplicaciones web?
Si
- ¿Desarrollarías un sistema web usando un Framework?
Si
- ¿Estarías dispuesto a leer o ver tutoriales para el aprendizaje de un Framework para aplicaciones web?
Si Por su puesto

Glosario

Aplicación Web: En la Ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un Servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación (Software) que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Base de datos: Es un conjunto de datos relacionados entre sí. Por datos entendemos hechos conocidos que pueden registrarse y que tienen significado implícito.

Lenguaje de programación: Un lenguaje de programación consiste en todos los símbolos, caracteres y reglas de uso que permiten a las personas "comunicarse" con las computadoras. Existen varios cientos de lenguajes y dialectos de programación diferentes. Algunos se crean para una aplicación especial, mientras que otros son herramientas de uso general más flexibles que son apropiadas para muchos tipos de aplicaciones. En todo caso los lenguajes de programación deben tener instrucciones que pertenecen a las categorías ya familiares de entrada/salida, cálculo/manipulación de textos, lógica/comparación y almacenamiento / recuperación.

Modularidad: Es el diseño de un sistema que se divide en un conjunto de unidades funcionales (llamadas módulos) de las que puede estar compuesta una aplicación más grande. Un módulo representa una serie de problemas relacionados. Se puede incluir un conjunto de componentes relacionados, tales como características, vistas o lógica de negocio y trozos de infraestructura. Los módulos son independientes uno del otro, pero pueden comunicarse entre sí de una manera imprecisa.

Navegador Web: Es un programa informático que facilita al usuario el navegar por internet, esto es, el poder visualizar e interactuar las páginas web de cualquier red. Su nombre viene de su capacidad para poder moverse, desplazarse de una página a otra mediante utilidades como los hipervínculos, que conectan las páginas situadas en distintos lugares del mundo entre sí con un solo click.

Programación artesanal: es una forma de mencionar a la programación que no tiene una estructura o no es organizada.

Servidor web: Es un programa que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un Navegador Web. Para la transmisión de todos estos datos se utiliza algún protocolo.

Los servidores de bases de datos: Es un sistema de gestión de base de datos que consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a ellos, es básicamente un sistema computarizado para llevar registros, es decir, un sistema cuya finalidad general es almacenar información y permitir a los usuarios realizar una variedad de operaciones sobre dichos registros, por ejemplo: insertar, borrar, modificar, agregar, consultar. La colección de datos se denomina base de datos.

WebForms: son los elementos de interfaz de usuario, que dan a las Aplicaciones Web de su apariencia. Web Forms son similares a Windows Forms ya que proporcionan propiedades, métodos y eventos para los controles que se colocan sobre ellos. Sin embargo, estos elementos de interfaz de usuario hacen así mismos en el marcador apropiado requerido por la petición, por ejemplo, HTML.

Referencias

- Ospina Torres, M., & Suárez Díaz, S. (2008). *Diseño y Desarrollo de una nueva Aplicación Web*. Caracas Venezuela.
- Alexander, C. (1977). *A Pattern Language. An inspiration for many people in the patterns movement. I'm less enamored of it than most, but it's worth*. Oxford: Oxford University Press.
- Atlassian. (2013). *Bitbucket Documentation*. Retrieved Mayo 30, 2013, from Bitbucket Documentation:
<https://confluence.atlassian.com/display/BITBUCKET/Bitbucket+Documentation+Home;jsessionid=3EBE1F147FF0B1093C62180DD6AC20C5>
- Blueprintcss. (2014, septiembre). Retrieved septiembre 2014, from Blueprintcss:
<http://www.blueprintcss.org/>
- Cake Software Foundation, Inc. (2014). *CakePHP*. Retrieved Septiembre 23, 2014, from CakePHP: <http://book.cakephp.org/2.0/es/index.html>
- Camargo Apen, M. A. (2008). *Técnicas para desarrollar E-Business exitosos, implementadas con técnicas basadas en AJAX*.
- Chacon, S. (2009). *Pro GIT* (Primera ed.). United States of America: Apress.
- Collins-Sussman, B., Fitzpatrick, B. W., & Pilato, C. (2004). *Version Control with Subversion*. O'Reilly Media.
- CVS HOME. (n.d.). *CVS HOME*. Retrieved Abril 19, 2013, from CVS HOME:
<http://www.cvshome.org/eng/>
- Django. (2014). Retrieved Septiembre 2014, from Django: <https://www.djangoproject.com/>
- Eguíluz Pérez, J. (2008). Introducción a AJAX. In J. Eguíluz Pérez, *introduccion a AJAX* (p. 251).
- EllisLab. (2013). *Codeigniter / Ellislab*. Retrieved Septiembre 25, 2014, from Codeigniter / Ellislab: <http://ellislab.com/codeigniter>
- Engine Yard. (2013). *About Engine Yard, Inc*. Retrieved Enero 14, 2013, from Engine Yard: <https://www.engineyard.com/company>
- Gutiérrez., J. J. (2014). *javierj@lsi.us.es*. Retrieved Septiembre 2014, from http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- JavaWorld. (2000). *HMVC: The layered pattern for developing strong client tiers*. Retrieved Septiembre 2014, from Javaworld:
<http://www.javaworld.com/javaworld/jw-07-2000/jw-0721-hmvc.html>
- Lazzarini, H. C. (2012). *Programacion Orientada a Objeto*.
- light, L. f. (2012, Septiembre). *Lighttpd*. Retrieved from <https://www.lighttpd.net/>
- MAMP. (2013). *MAMP: Mac - Apache - MySQL*. Retrieved Mayo 31, 2013, from MAMP: Mac - Apache - MySQL: <http://www.mamp.info/en/mamp/index.html>
- manual.html. (2014). Retrieved Septiembre 25, 2014, from html: <http://www-app.etsit.upm.es/~alvaro/manual/manual.html>
- mongrel2. (2011). *mongrel2*. Retrieved agosto 2015, from <http://mongrel2.org/>
- MySQL. (2011). *MySQL :: MySQL 5.0 Reference Manual*. Retrieved Septiembre 25, 2014, from MySQL Dev: <http://dev.mysql.com/doc/refman/5.0/es/what-is.html>

Oracle Corporation and/or its affiliates. (2012). *Netbeans.org*. Retrieved Noviembre 08, 2012, from Netbeans.org: http://netbeans.org/index_es.html

passenger, P. (2012). *Phusion passenger*. Retrieved agosto 2015, from Phusion passenger: <https://www.phusionpassenger.com/>

Rails. (2013). *Rails*. Retrieved agosto 2015, from Ruby on Rails: <http://rubyonrails.org/>

Ramírez, A. T. (2014). *Sistema de Administración de Documentos Corporativos Digitales*. Queretaro.

Ruby Lang Org. (2013). *Lenguaje de Programación Ruby*. Retrieved Marzo 8, 2013, from Lenguaje de Programación Ruby: <http://www.ruby-lang.org/es>

The PHP Documentation Group. (2012). *Prefacio*. Retrieved Septiembre 25, 2014, from PHP: <http://www.php.net/manual/es/preface.php>

Vazquez Esteban, D. (2013). *Sistema de Administracion y Monitoreo utilizando dispositivos moviles*. tuxtla gutierrez.

W3 org. (2002, Marzo 4). *W3/Protocols*. Retrieved Septiembre 15, 2014, from W3/Protocols: <http://www.w3.org/Protocols/Specs.html>

W3 org. (2014). *W3 - CSS*. Retrieved Septiembre 2014, from W3 - CSS: <http://www.w3.org/Style/CSS/>

WampServer. (2014). *WampServer*. Retrieved Septiembre 25, 2014, from WampServer: <http://www.wampserver.com/en/>

WEBrick. (2013). *WEBrick*. Retrieved agosto 2015, from WEBrick: <http://ruby-doc.org/stdlib-2.0.0/libdoc/webrick/rdoc/WEBrick.html>

XAMPP. (2015). Retrieved marzo 2015, from XAMPP Installers and Downloads: <https://www.apachefriends.org/es/index.html>

Zend. (2014). *Zend Framework & MVC Introduccion - Zend Framework Quick Start*. Retrieved Septiembre 25, 2014, from Zend Framework: <http://framework.zend.com/manual/1.12/en/learning.quickstart.intro.html#learning.quickstart.intro.mvc>

Zend Framework. (2014). *Zend Framework*. Retrieved Septiembre 25, 2014, from Zend Framework: <http://framework.zend.com/manual/1.12/en/introduction.overview.html>