



ISBN: 978-99961-50-06-7

ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA – FEPADE
DIRECCIÓN DE INVESTIGACIÓN Y PROYECCIÓN SOCIAL

PROGRAMA DE INVESTIGACIÓN APLICADA

INFORME FINAL DE INVESTIGACIÓN

**“CREACIÓN DE UN FRAMEWORK DE DESARROLLO DE
APLICACIONES PARA LA WEB CON LENGUAJE PHP”**

SEDES Y ESCUELAS PARTICIPANTES: CENTRO REGIONAL MEGATEC ZACATECOLUCA

DOCENTE INVESTIGADOR RESPONSABLE: TÉC. JOSÉ ANTONIO HENRÍQUEZ CHAVARRÍA

DOCENTES INVESTIGADORES PARTICIPANTES: LIC. SANTIAGO EDUARDO DOMÍNGUEZ
TÉC. LUIS ALEXANDER MÉNDEZ PALACIOS
ING. ANA CECILIA ÁLVAREZ DE VENTURA

SANTA TECLA, ENERO 2013



ISBN: 978-99961-50-06-7

**ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA – FEPADE
DIRECCIÓN DE INVESTIGACIÓN Y PROYECCIÓN SOCIAL**

PROGRAMA DE INVESTIGACIÓN APLICADA

INFORME FINAL DE INVESTIGACIÓN

**“CREACIÓN DE UN FRAMEWORK DE DESARROLLO DE
APLICACIONES PARA LA WEB CON LENGUAJE PHP”**

SEDES Y ESCUELAS PARTICIPANTES: CENTRO REGIONAL MEGATEC ZACATECOLUCA

DOCENTE INVESTIGADOR RESPONSABLE: TÉC. JOSÉ ANTONIO HENRÍQUEZ CHAVARRÍA

DOCENTES INVESTIGADORES PARTICIPANTES: LIC. SANTIAGO EDUARDO DOMÍNGUEZ
TÉC. LUIS ALEXANDER MÉNDEZ PALACIOS
ING. ANA CECILIA ÁLVAREZ DE VENTURA

SANTA TECLA, ENERO 2013

Rectora
Licda. Elsy Escolar SantoDomingo
Vicerrector Académico
Ing. José Armando Oliva Muñoz
Vicerrectora Técnica Administrativa
Inga. Frineé Violeta Castillo

Dirección de Investigación y Proyección Social
Ing. Mario Wilfredo Montes
Ing. David Emmanuel Agreda
Lic. Ernesto José Andrade
Sra. Edith Cardoza

Director Coordinador del Proyecto
Ing. René Edgardo Flores Monroy

Autores
Téc. José Antonio Henríquez Chavarría
Lic. Santiago Eduardo Domínguez

005.3
H467

sv
Henríquez Chavarría, José Antonio
Creación de un Framework de desarrollo de aplicaciones para la web con lenguaje PHP /
José Antonio Henríquez Chavarría, Santiago Eduardo Domínguez. -- 1ª ed. --
San Salvador, El Salvador: ITCA Editores, 2013.

35 p.: il. ; 28 cm.
ISBN: 978-99961-50-06-7

1. Programación (Computadores electrónicos). 2. Framework-Programa de computador.
I Domínguez, Santiago Eduardo, coaut. II. Escuela Especializada en Ingeniería ITCA-FEPADE.



El Documento Creación de un framework de desarrollo de aplicaciones para la web con lenguaje PHP es una publicación de la Escuela Especializada en Ingeniería ITCA – FEPADE. Ha sido concebido para difundirlo entre la comunidad académica y el sector empresarial, como un aporte al desarrollo del país. El contenido de la investigación puede ser reproducida parcial o totalmente, previa autorización escrita de la Escuela Especializada en Ingeniería ITCA–FEPADE. Para referirse al contenido, debe citar la fuente de información. El contenido de este documento es responsabilidad de los autores.

Sitio web: www.itca.edu.sv
Correo electrónico: bibliotecologos@itca.edu.sv
Tiraje: 16 ejemplares
PBX: (503) 2132 – 7400
FAX: (503) 2132 – 7423
ISBN: 978-99961-50-06-7
Año 2013

ÍNDICE

CONTENIDO	PÁGINA
ÍNDICE	3
1. INTRODUCCIÓN.....	4
2. PLANTEAMIENTO DEL PROBLEMA.....	4
2.1 DEFINICIÓN DEL PROBLEMA	4
2.2 JUSTIFICACIÓN	5
2.3 OBJETIVOS	5
2.3.1 OBJETIVO GENERAL	5
2.3.2 OBJETIVOS ESPECÍFICOS:	5
3. ANTECEDENTES	6
4. MARCO TEÓRICO DE LA INVESTIGACIÓN	7
5. METODOLOGÍA DE LA INVESTIGACIÓN.	9
5.1 ETAPA 1: DETERMINACIÓN DEL PROBLEMA	9
5.2 ETAPA 2: ANÁLISIS DEL PROBLEMA.	9
5.3 ETAPA 3: DISEÑO DE LA APLICACIÓN	10
5.4 ETAPA 4: DESARROLLO.....	10
5.5 ETAPA 5: DOCUMENTACIÓN Y PUBLICACIÓN.	10
5.6 HERRAMIENTAS Y METODOLOGÍAS IMPLEMENTADAS EN LA CONSTRUCCIÓN.	11
5.7 DIAGRAMAS UML	15
5.8 TIPO DE LICENCIA:.....	20
5.9 CONFIGURACIÓN Y PREPARACIÓN DEL ENTORNO.....	20
5.10 PASOS PARA GENERAR UNA APLICACIÓN UTILIZANDO EL FRAMEWORK (GUÍA DE USUARIO).	21
5.11 RECOMENDACIONES PARA MODIFICAR EL CÓDIGO DE LA APLICACIÓN GENERADA:	27
6. CONCLUSIONES.....	28
7. RECOMENDACIONES.	28
8. REFERENCIAS:	29
9. ANEXOS.....	29

1. INTRODUCCIÓN

El desarrollo de Frameworks, para desarrollo de aplicaciones en ambiente web, ha tenido un notable aumento, en los últimos años, debido a la necesidad de crear aplicaciones web y la urgencia de muchos por conseguir desarrollos con un ciclo de vida rápido, de forma que actualmente muchos desarrolladores de software requieren herramientas como un Framework para lograr programar sus aplicaciones y dar seguimiento para siguientes versiones.

El presente proyecto se plantea como una alternativa salvadoreña para el desarrollo de software, en entornos web y utilizando PHP como lenguaje de desarrollo. Esto con el propósito de facilitar el desarrollo de software web a bajos costos y el acceso a empresas pequeñas y medianas a las alternativas de código libre.

2. PLANTEAMIENTO DEL PROBLEMA

2.1 DEFINICIÓN DEL PROBLEMA

Un Framework es una herramienta base para la generación de una aplicación utilizando lenguajes y entornos de desarrollo específicos.

El desarrollo de aplicaciones Web con frecuencia, se vuelve una tarea rutinaria que trae consigo el crear muchos archivos, los cuales son fundamentales para el funcionamiento de la aplicación pero que retrasan el tiempo para el desarrollo de la aplicación, que son similares y solamente se realizan cambios mínimos lo cual es para los desarrolladores un trabajo tedioso y que quita tiempo para enfocarse en la lógica real de la aplicación

Los Framework debe permiten automatizar la creación de los archivos comunes de toda aplicación como lo son los formularios, los procesos de guardar, modificar, consultar y eliminar registros de una Base de Datos; ya que éstos son los procesos en que los desarrolladores toman más tiempo, con la creación de la herramienta se implementarán nuevas metodologías de desarrollo de aplicaciones como lo son el modelo de tres capas MVC (modelo vista-controlador) y el uso de ORM(mapeo relacional de objetos) los cuales ayudará a ser más productivos a la hora de crear aplicaciones Web.

2.2 JUSTIFICACIÓN

En El Salvador, existe una comunidad de programadores que utiliza principalmente herramientas gráficas de desarrollo web como Dreamweaver y Netbeans, para aumentar su productividad, sin embargo la generación de código que facilite el mantenimiento de una base de datos y un software base para completar y dar mantenimiento a la aplicación web no existe sin utilizar un Framework. Los Framework son aplicaciones que brindan un entorno con un conjunto de librerías para facilitar el desarrollo de aplicaciones. Actualmente existen muchos, la mayoría de los cuales se han vuelto herramientas para desarrolladores avanzados y especializados en cada uno de ellos.

La necesidad de un Framework para desarrolladores con conocimiento intermedios y básicos de programación con PHP es por lo tanto necesaria, ya que se convierte en una herramienta que facilitará y promoverá el desarrollo de software de código abierto y aumentaría el acceso a las empresas pequeñas y medianas a desarrollos de sistemas livianos con bajos costos y a su alcance.

2.3 OBJETIVOS

2.3.1 OBJETIVO GENERAL

Desarrollar un Framework para el desarrollo de aplicaciones con PHP, que facilite la creación de software a empresas con el fin de mejorar los procesos en sus operaciones.

2.3.2 OBJETIVOS ESPECÍFICOS:

- Crear un Framework para desarrollo web con software libre utilizando como lenguaje PHP.
- Proveer una herramienta a programadores que permita el desarrollo rápido de aplicaciones web en un entorno de software libre.
- Brindar la documentación necesaria para que los programadores web utilicen libremente el Framework desarrollado en la programación de aplicaciones en línea.

3. ANTECEDENTES

En El Salvador, el desarrollo de software es una industria que inicia, con esfuerzos de un grupo reducido de empresas y profesionales que se dedican al desarrollo de sistemas o aplicaciones a la medida de empresas medianas y grandes, principalmente.

En la comunidad de programadores en el Salvador no se ha encontrado evidencia que se haya desarrollado Framework para desarrollo web, de hecho la mayoría de programadores que conocen los frameworks para aplicaciones web son programadores a nivel avanzado de PHP y los utilizan solamente para manejar o desarrollar sistemas grandes.

La palabra inglesa "Framework" define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

Actualmente existen muchos Framework desarrollados entre los más usados tenemos:

- **Zend Framework**
- **CakePHP**
- **Symfony**
- **CodeIgniter**
- **YiiFramework**

Todos con muchas características potentes para desarrollo de aplicaciones web, sin embargo estos requieren un manejo avanzado de lenguaje PHP para utilizarlos en el desarrollo rápido de aplicaciones. Por lo que se han convertido en herramientas para desarrolladores avanzados principalmente utilizadas en proyectos grandes.

En el desarrollo de software, un Framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

4. MARCO TEÓRICO DE LA INVESTIGACIÓN

4.1 DEFINICIONES GENERALES:

FRAMEWORK (Entorno de Trabajo): Un entorno que facilita el desarrollo de portales, aplicaciones grandes y escalables. Buscan también facilitar el manejo y desarrollo de proyectos grandes. Los Frameworks de PHP permiten principalmente generar objetos de acceso a datos, crear una estructura de clases para el código y brindar una estructura física para el proyecto en desarrollo, que facilita el mantenimiento del código generado.

- **MVC:** Modelo Vista Controlador. Donde el **controlador** es el encargado de controlar el acceso a la aplicación o cualquier aspecto de la misma, es el corazón del nuestro Framework. Recibe una petición, la analiza, recaba datos de la Base de Datos busca la vista asociada y muestra los resultados; el **modelo**: Es un componente encargado de las operaciones lógicas y de los datos, es quien se conecta con la base de datos y recupera resultados; y el componente **vista** es el que se encarga de presentar utilizando plantillas de diseño los datos obtenidos, en otras palabras es la interfaz gráfica de la aplicación.

Gráficamente el Modelo Vista Controlador se representa así:

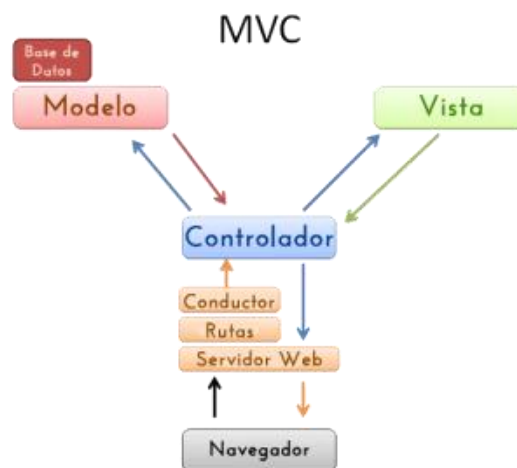


Fig. 1: El Modelo Vista Controlador

- **PDO (PHP Data Objects):** Es una extensión que provee una capa de abstracción de acceso a datos para PHP 5, con lo cual se consigue hacer uso de las mismas funciones para hacer consultas y obtener datos de distintos manejadores de bases de datos.

- **Programación Orientada a Objetos:** Es un paradigma de programación que usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo, y encapsulamiento.
- **CSS y CSS3:** El nombre hojas de estilo en cascada viene del inglés *Cascading Style Sheets*, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).
- **El W3C** (World Wide Web Consortium): es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.
- **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 y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. Al ser un software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.
- **DOM:** El Document Object Model o DOM ('Modelo de Objetos del Documento' o 'Modelo en Objetos para la representación de Documentos') es esencialmente una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML
- **PHP:** Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (*server-side scripting*) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas
- **MySQL:** Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 es una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation; desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

4.2 LOS FRAMEWORK:

Los Framework se caracterizan principalmente por:

- Brindar un entorno para desarrollo planificado e integrado para el desarrollador
- Facilitar el mantenimiento y actualización del código fuente de la aplicación
- Mediante su estructura brindan ¹ :
 - Funciones de uso común / objetos.
 - Una lógica de negocio.
 - Un diseño base para la aplicación
- Aplican técnicas de programación actualizadas y estandarizadas.

Lo anterior permite brindar cierta garantía de soporte y ventaja para el desarrollo rápido al programador, que busca minimizar los tiempos y garantizar que el software en desarrollo sea eficiente, escalable y configurable.

5. METODOLOGÍA DE LA INVESTIGACIÓN.

La investigación aplicada tiene como principal objetivo, resolver problemas prácticos; por lo tanto se toma de base para la metodología de la presente investigación, con el objetivo que los resultados, sean conforme a las necesidades de las microempresas de la Región de Los Nonualcos.

5.1 ETAPA 1: DETERMINACIÓN DEL PROBLEMA

En esta etapa se determinaron las diversas alternativas para desarrollar el software Framework, analizando el estado actual de Frameworks existentes y demás aplicaciones.,

También se analizó los requerimientos que todo programador necesita para desarrollar una aplicación web utilizando bases de datos.

5.2 ETAPA 2: ANÁLISIS DEL PROBLEMA.

En esta etapa se determinó los requerimientos del software a desarrollar y los componentes que manejaría, así como se analiza las necesidades del programador y alternativas para el diseño y desarrollo.

¹ Romero Alex, URL:<http://blog.suenyos.com/2011/07/20/crear-un-framework-en-php5-desde-cero-1%C2%AA-parte/>

5.3 ETAPA 3: DISEÑO DE LA APLICACIÓN

El diseño de la aplicación consiste en definir los estándares y estructura de la aplicación, se define las estructuras conceptuales, lógicas y físicas de la aplicación utilizando en este caso el MVC como estándar guía para diseñar y estructurar la aplicación.

5.4 ETAPA 4: DESARROLLO.

Se aplican y utilizan herramientas de desarrollo y lenguaje de programación creando clases y estructuras de datos que permitan la generación de aplicaciones en PHP.

5.5 ETAPA 5: DOCUMENTACIÓN Y PUBLICACIÓN.

Se documenta la aplicación para los usuarios, además se espera publicar en un sitio web la aplicación para que programadores puedan utilizarla, y evaluarla; para generar futuros desarrollos o versiones del mismo.

ANÁLISIS DE REQUERIMIENTOS

Resumen de procesos analizados para el funcionamiento del Framework.

1. Generación de formularios por cada tabla de una base de datos.

Este proceso es básico para poder desarrollar todas las funciones adicionales del Framework, ya que con esto generamos un mapeo de la base de datos y podemos saber los nombres de campos, tipo de datos y longitud de los datos de cada tabla, lo cual permite el poder controlar la generación y automatización de las funciones que serán base para el funcionamiento de la herramienta.

2. Validación de entrada de datos de formularios.

Los formularios creados automáticamente deberán poder controlar los datos que estos procesaran, en este caso se utilizará la validación con una regla básica que todos los campos deberán de contener datos.

3. Generación Automática de modelos de datos.

Los modelos de datos son las especificaciones de como asociar los datos de cada tabla de la base de datos en una forma de objeto, para así poder generar el encapsulamiento, herencia, polimorfismo y, poder trabajar sobre una metodología de programación orientada a objetos.

4. Generación de controladores para las funciones de mantenimiento de los datos de las tablas.

Las funciones básicas de consulta, ingreso, actualización y borrado de registros son primordiales para todo sistema que maneja transacciones y que en un periodo de tiempo estos registros generan reportes para la toma de decisiones.

5. Generación de páginas con resultado de consultas.

Estos componentes serán nuestra salida básica de información y así los usuarios sepan los movimientos e información con la que cuenta sus aplicativos.

6. Creación de controles dinámicos.

Se necesitan componentes específicos y que cumplan funciones específicas del manejo de datos, por ello se toma en consideración la implementación de los mismos y serán parte fundamental de las interfaces de captura de información.

5.6 HERRAMIENTAS Y METODOLOGÍAS IMPLEMENTADAS EN LA CONSTRUCCIÓN.

En el análisis de las herramientas necesarias para la creación del Framework, se opta por utilizar herramientas de código libre para su desarrollo, entre los factores que nos permiten decidir por dichas herramientas están:

- Son de fácil acceso para la mayoría de usuarios.
- Amplia documentación y soporte por las comunidades de desarrollo.}
- Multiplataforma.

Herramientas para el desarrollo.

En el desarrollo se han utilizado 2 herramientas las cuales nos permiten el fácil uso de depuradores, librerías y documentación del lenguaje de programación que se está utilizando. Se ha utilizado dos editores uno para el trabajo directo de codificación y otro para realizar las pruebas con el sistema gestor de base de datos, dichas herramientas son:

- Netbeans IDE 7.2.²
- MySQL Workbench 5.2.³

La opción de uso de Netbeans IDE es por fácil uso, integración con varios lenguajes de programación, incluyendo PHP que es la opción seleccionada para la codificación del

² Amplia descripción del software en: <http://netbeans.org/>

³ Amplia descripción del software en: <http://www.mysql.com/products/workbench/>

Framework, integración con plugins y extensiones para mejorar su uso y herramientas de depuración y reporte de fallos en la codificación.

Por otra parte el uso de MySQL Workbench, es por su gran integración con el sistema gestor de base de datos MySQL el cual es el que se utiliza para hacer las pruebas correspondientes del manejo de datos que el Framework utilizará en el desarrollo y pruebas.

Entorno de desarrollo y pruebas.

Para el desarrollo del Framework se utiliza un paquete de herramientas de código libre las cuales nos permiten implementar los diferentes servidores y servicios, con lo que se realizan las pruebas y codificación de la aplicación, este paquete es:

WampServer Versión 2.2⁴ para sistemas operativo Microsoft Windows de 32 bit⁵, el cual integra en su entorno las siguientes herramientas:

- Interprete de lenguaje: PHP 5.4.3
- Servidor Web: Apache 2.2.22
- Sistema gestor de base de datos: MySQL 5.5.24.

Otras tecnologías y herramientas que se han tomado en cuenta para el desarrollo son lenguajes de presentación de contenido en la Web, las cuales hoy en día se han vuelto un estándar para su uso en diferentes entornos no solo Web sino en ambientes móviles y de trabajo en plataformas de escritorio, se utilizan los estándares de HTML5 y CSS3⁶, para el desarrollo de interfaces de captura de datos, salidas de datos y diseño de todos los componentes y elementos visuales que el Framework creará, o los cuales utilizará para presentación de información al usuario final, además se utiliza el lenguaje JavaScript para poder tener una integración con los clientes web y así poder manejar ciertas acciones de una manera más óptima y rápida, las cuales realizándolas solo con PHP retardan los resultados esperados por el cliente final.

Por la utilización de CSS y JavaScript se han utilizado los siguientes componentes los cuales facilitan el trabajo del diseño visual de las aplicaciones creadas desde el Framework, estas son:

⁴ Amplia descripción del software en: <http://www.wampserver.com/>.

⁵ Se han utilizado 2 sistemas operativos, una distribución de Ubuntu basada en Linux, pero el mayor desarrollo se realiza en el SO de Microsoft.

⁶ Amplia información en: <http://www.w3.org/standards/webdesign/htmlcss>.

- Librería jquery-1.8.2, permite un estándar de uso del JavaScript tradicional, implementando una característica es compatibilidad con la mayoría de navegadores en el mercado⁷.
- Librería jquery-ui-1.8.16, librería de componentes para la generación de interfaces visuales como calendarios, selectores de tiempo, paneles y pestañas.
- Librería jquery.validate, plugin de jquery para la validación de datos de formularios con muchas características y reglas de validación para la integridad de los datos que se utilizaran⁸.
- Librería modernizr-2.6.1-respond-1.1.0(implementación de boilerplate), paquete de librería para poder implementar elementos de HTML5 y CSS3 que no son soportado por navegadores antiguos, además de la inclusión del componente respond para el diseño web adaptable (responsive design) en otras palabras adaptar el diseño a diferentes formatos de presentación.
- Boilerplate 4.1, esta es una plantilla de diseño, proporcionados para la implementación de HTML5 y CCS3, es adaptable y cumple con los estándares de implementación para la mayoría de navegadores⁹.
- Librería bootstrap 2.2.2¹⁰, Framework creado por twitter para la mejor implementación de estilos y simplificación de la aplicación de los mismos para un rápido desarrollo de interfaces y compatibilidad en el diseño, esta nueva versión implementa una librería completa de media query's las cuales nos ayudan al diseño adaptable en los mayoría de dispositivos.

Todos estos componentes permiten la implementación de interfaces adaptables, complementan el diseño de los formularios y páginas web que en el Framework son las que nos sirven para presentar o capturar datos.

Con todo los compontes antes mencionado más la programación del lado del servidor se puede desarrollar todos los procesos descritos en los requerimientos, ahora en el desarrollo y codificación del Framework se ha necesitado investigar que técnicas son las mejores que podemos utilizar para poder desarrollar una herramienta rápida, flexible, escalable y segura para el uso e implementación, las técnicas utilizadas han sido las siguientes.

⁷ Características de jquery : <http://es.wikipedia.org/wiki/JQuery#Caracter.C3.ADsticas>.

⁸ Amplia información del plugin: <http://bassistance.de/jquery-plugins/jquery-plugin-validation/>

⁹ Amplia información <http://html5boilerplate.com/>.

¹⁰ Información y Documentación de bootstrap: <http://twitter.github.com/bootstrap/>

1. Programación Orientada a Objetos.

Se ha optado al desarrollo utilizando esta técnica de programación para mejorar la comprensión y uso de cada elemento creado para el Framework ya que nos podemos acercar más a la realidad y simpleza del uso de elementos con características y funciones además de aprovechar los mecanismos que la POO implementa (Herencia, polimorfismo, encapsulamiento) para un mejor rendimiento y aprovechamiento de los recursos de los programas creados con esta técnica.

2. Patrones de Diseño (Design patterns)¹¹.

Estos son mecanismos de uso de clases y objetos que se repiten y dan solución a problemas comunes en el desarrollo de software.

3. Modelo Vista-Controlador (MVC).

Este es una técnica de diseño dirigida a la separación de los elementos de codificación según su uso, en este caso se categorizan por la función que cumplen dentro de nuestra aplicación en este caso las categorías son: el tratamiento de datos, las funciones y control del flujo de información entre los componentes, la captura de datos y muestra de resultados.

¹¹ Objetivo del uso de patrones de diseño http://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o#Objetivos_de_los_patrones

DISEÑO DE APLICACIÓN.

5.7 DIAGRAMAS UML

Casos de USO.

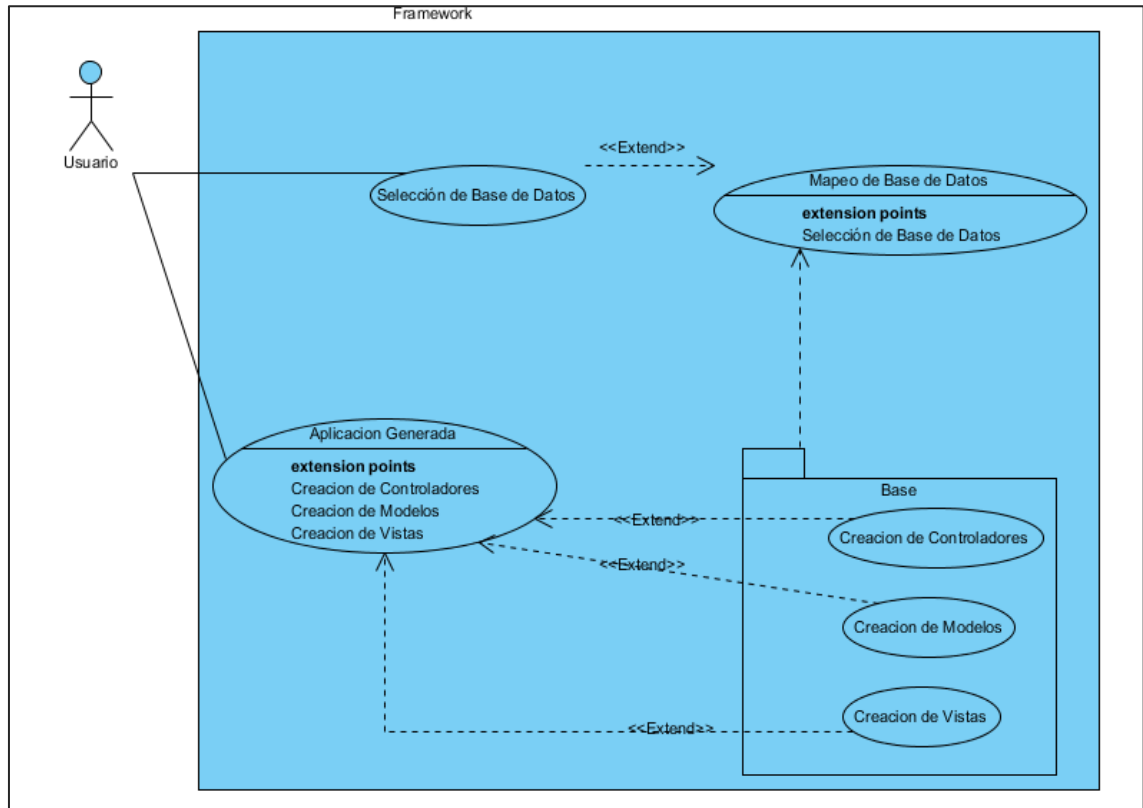


Fig.2. Caso de uso general del Framework

Como punto de partida para la creación de aplicaciones con el Framework, es el de tener una base de datos, ya que esta proporciona la información para la creación de cada componente y estructura de la aplicación, su aplica una lógica del negocio en su aspecto más general.

Explicación de componentes.









Nombre	Documentación
 Framework	Ámbito de la aplicación en su máximo estado o nivel
 Base	Paquete con las funcionalidades básicas dentro del Framework
 Aplicación Generada	Suceso que se da cuando se crea la aplicación solicitada por el actor
 Mapeo de Base de Datos	Proceso que lee, clasifica y ordena la forma de cómo se crearan los componentes, formularios y archivos de la aplicación
 Creación de Vistas	Caso o suceso que auxilia al mapeo para generar los elementos de captura de datos y presentación de resultados
 Creación de Modelos	Suceso que auxilia al mapeo para la creación de los modelos que permitirán el control de los datos que se procesaran en la aplicación.
 Creación de Controladores	Suceso que auxilia al mapeo para poder crear las funciones que permitirán el mantenimiento de los datos, la muestra de resultados y otras acciones generales de cada aplicación.
 Selección de Base de Datos	Caso en donde se decide la base de datos que se utilizará para poder generar la aplicación automáticamente

Tabla.1. Componentes del Framework

Como se puede apreciar los casos generales son el de selección de base de datos y la aplicación generada los demás elementos complementan cada caso y se analizaran más adelante en sus componentes internos.

Diagrama de clases.

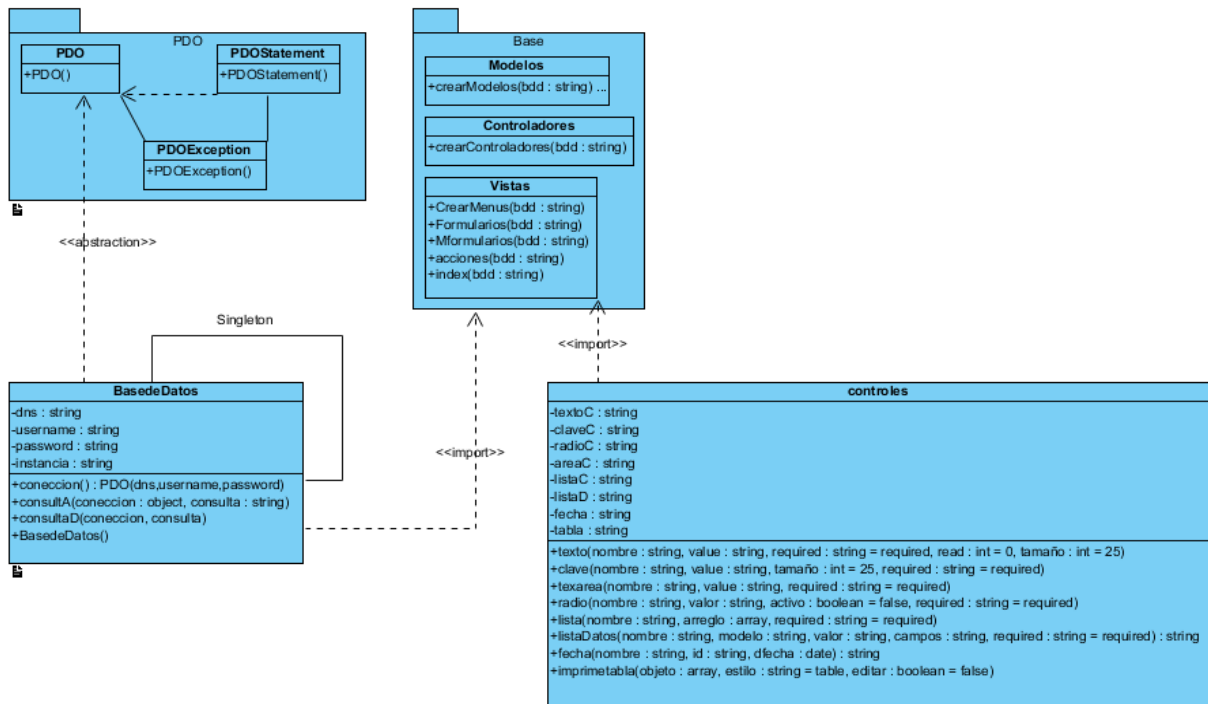


Fig. 3. Diagrama de Clases

Resumen











Nombre	Documentación
 PDO	API de implementación de uso de conexión a objetos de datos de PHP es una nueva implementación para la estandarización del uso de conexiones a base de datos
 Base	Paquete para las clases base que generan todo los componentes y/o archivos para el funcionamiento de las aplicaciones
 PDO	Clase que implementa una capa de abstracción de acceso datos independientes del sistema gestor de base de datos. Representa una conexión entre PHP y un servidor de bases de datos.
 PDOStatement	Clase que permite la utilización de sentencias preparadas, para dar paso a las ejecuciones de las instrucciones a la base de datos y además generar un resultado de datos asociado.
 PDOException	Clase que se encarga de controlar las excepciones generadas por las clases del API.
 Modelos	Clase encargada de revisar los atributos de las tablas de la base de datos y generar un archivo o una clase modelo con los atributos encapsulados para un manejo más adecuado en la programación orientado a objetos, genera los métodos set y get para cada atributo.
 Controladores	Clase que se encarga de generar archivos con las clases que implementan los métodos para el manejo de datos como guardar, modificar borrar y consultar.
 Vistas	Clase que genera los archivos necesarios para las interfaces de captura de datos, muestra de resultados y manejo de acciones de acuerdo a las opciones que el usuario realice cuando se ejecuten las aplicaciones.
 BasedeDatos	Clase para el manejo de las conexiones a la base de datos hace uso de la clase PDO, se utiliza un patrón singleton para poder mantener una única conexión por petición.
 controles	Clase que se encarga de implementar y manejar los controles dinámicos para las interfaces de captura de datos, los cuales implementa validación automática de los datos, relaciones entre datos y calendarios.]

Tabla 2. Resumen de las clases de la aplicación

Diagrama de secuencias.

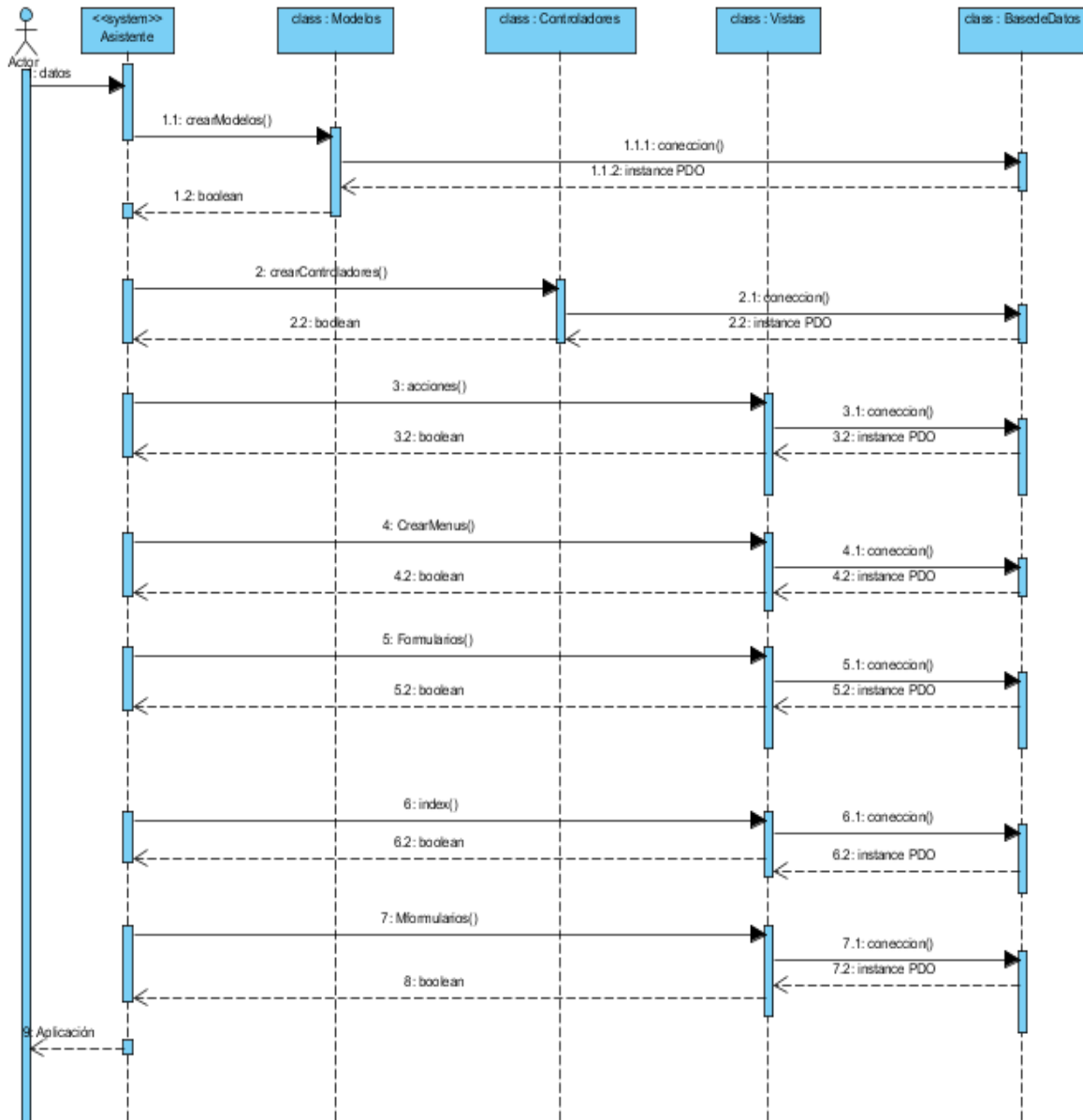


Fig. 4. Diagrama de Secuencias

Diagrama de comunicación.

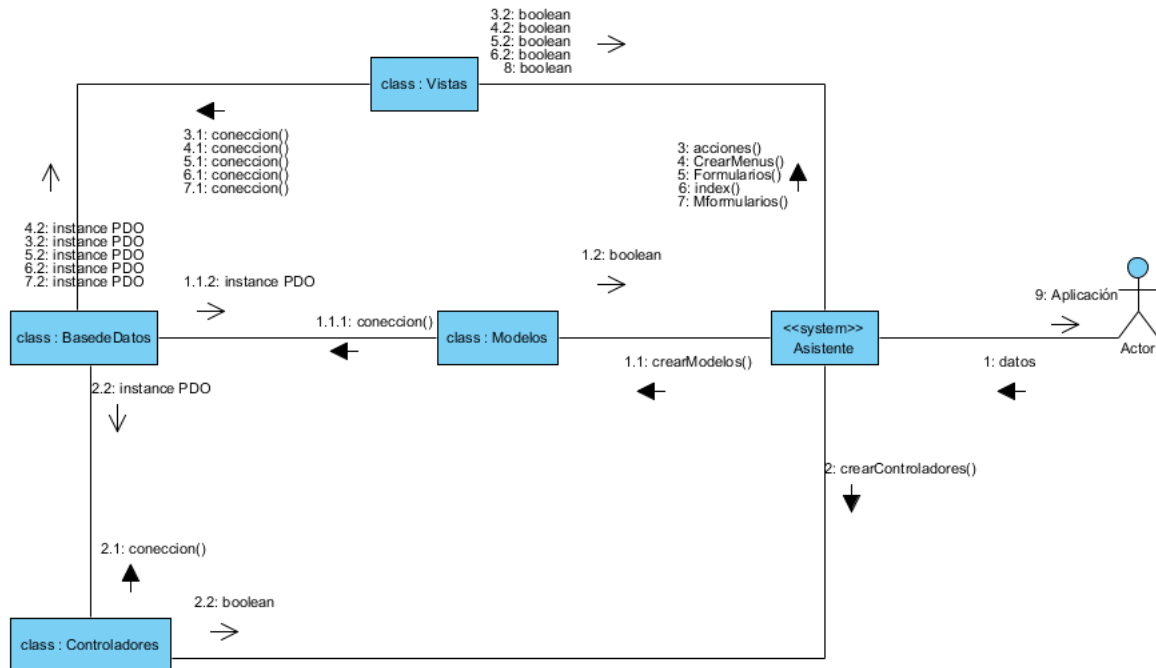


Fig. 5. Diagrama de Comunicación

5.8 TIPO DE LICENCIA:

Se consideró publicar el software bajo los términos de Licencia de **Apache.org**: (*Versión 2.0, enero 2004*), los cuales permiten a los usuarios realizar modificaciones que considere, comercializar o continuar dejando libre de uso los productos de software generados con base al presente Framework.

Puede ver los detalles de la licencia en **el Anexo 1**.

5.9 CONFIGURACIÓN Y PREPARACIÓN DEL ENTORNO

Para la correcta ejecución del Framework se deben de cumplir los siguientes requisitos.

- Tener instalado y configurado un entorno web con manejo de base de datos MySQL e intérprete de lenguaje PHP.
- Tener una base de datos que cumpla las siguientes reglas, debe de haber por lo menos una tabla, todas las tablas deben de poseer un campo con nombre "id" y debe

de ser del tipo numérico entero, con la característica de autoincrementado; todas las tablas deben de seguir los nombres según las convenciones de nombre.

- Crear una carpeta en el servidor web y copiar los archivos base del framework la estructura de archivos queda como muestra la siguiente imagen.

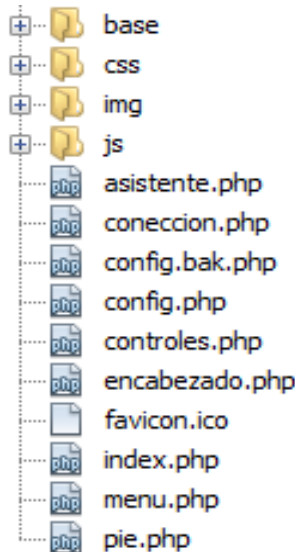


Fig. 6. Estructura de archivos del framework

- La carpeta base debe de tener permisos de escritura.

Estas son las características que debe de tener el entorno para poder ejecutar el framework, el no tomar en cuenta alguna de las mismas influye en el mal funcionamiento o ejecución inadecuada de algunos de los scripts que se necesitan para el funcionamiento óptimo de la aplicación

5.10 PASOS PARA GENERAR UNA APLICACIÓN UTILIZANDO EL FRAMEWORK (GUÍA DE USUARIO).

1. Configurar utilizando config.php:

Para poder generar una aplicación con el framework se debe de considerar los pasos anteriores, al cumplir con estos requisitos podemos realizar una configuración básica en el archivo config.php para poder asignar algunos datos básicos para la configuración del entorno de ejecución de los archivos del framework esta es la estructura del archivo config.php.

```
1 <?php
2 define("BASE", "");
3 define("NOMBRE", "");
4 define("TITULO", "");
5 define("DBNAME", "");
6 define("CLAVE", "");
7 define("HOST", "");
8 define("USUARIO", "");
9 define("URL", "");
10 define('DNS', '');
11 ?>
```

Fig. 7. Configuración de variables de entorno.

Las líneas que se deben de editar serán las 5,6,7 y 8 estas dependen de las configuraciones que tenga el sistema gestor de base de datos, la línea 5 es el nombre de una base de datos que este creada previamente en el sistema se recomienda usar la por default que es “mysql”, las otras líneas se configuran automáticamente después de ejecutar el asistente.

2. Definiendo la información para el servidor de base de datos y aplicación a generar:

El siguiente paso es ejecutar en el navegador web de la preferencia, es acceder a la URL del servidor local y acceder a la carpeta que se creó y donde se copiaron los archivos del framework, ejemplo:

<http://localhost/framework/>

Entonces accederemos a la siguiente página que se muestra en la imagen, la cual es donde se configuran todas las características para la ejecución del proyecto así como elegir la base de datos con la cual se generara la nueva aplicación.

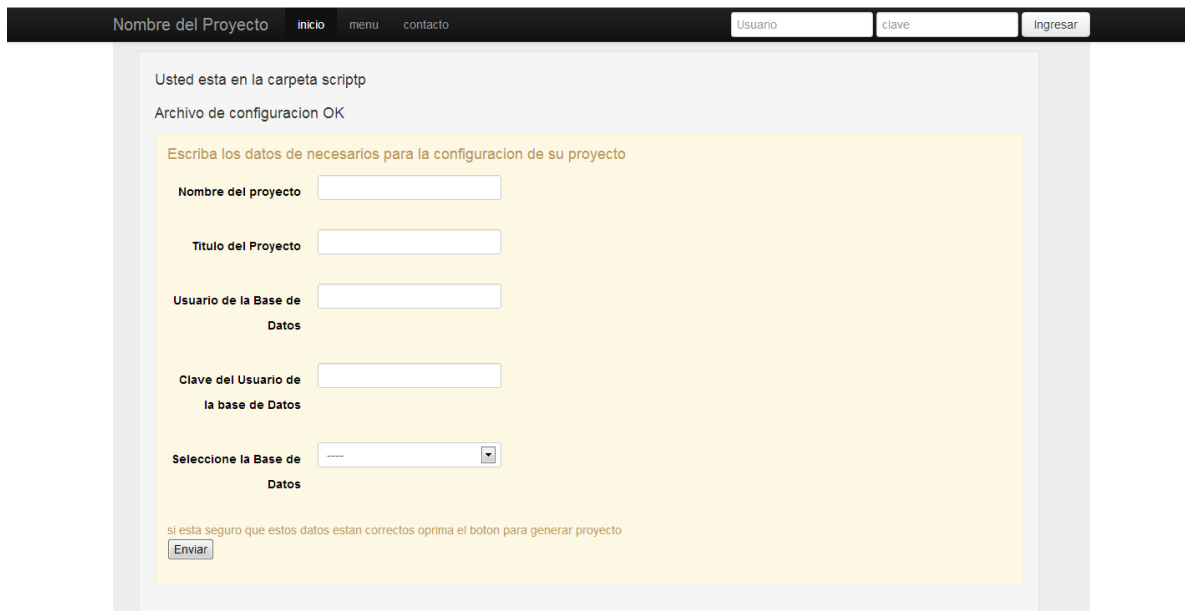


Fig. 8. Pantalla inicial para de creación de la aplicación

Al llenar los datos que se solicitan y seleccionar la base de datos de la aplicación que se generará, se crearan 3 carpetas las cuales son modelos, controladores y vistas con archivos que son equivalentes a cada tabla de la base de datos seleccionada, en estas imágenes vemos la estructura creada en el proyecto.

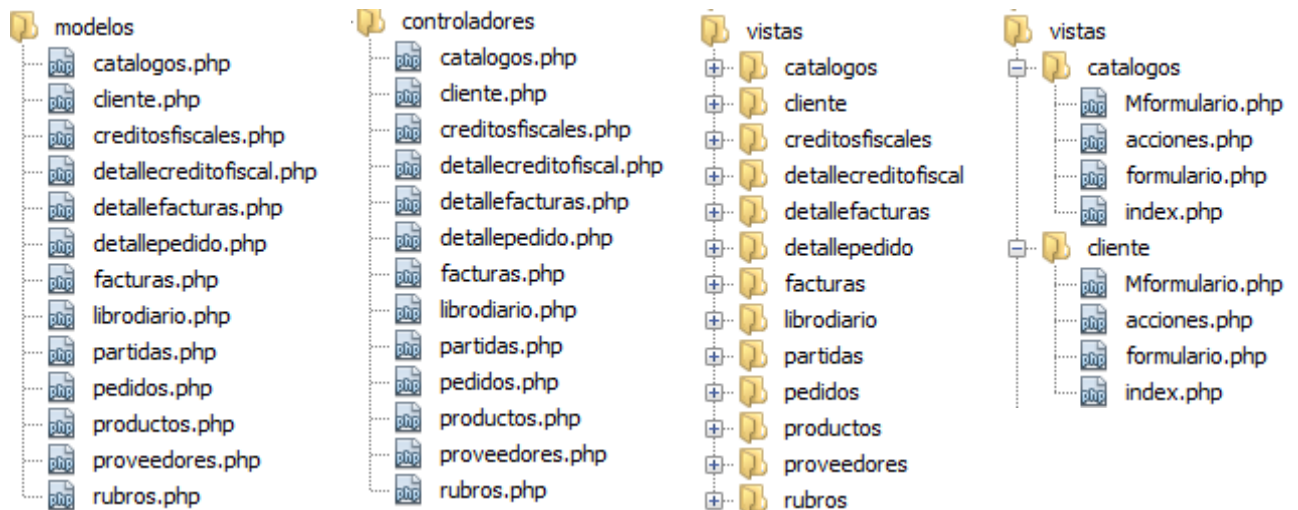


Fig. 9. Estructura de archivos generados.

Como podemos observar esta nueva estructura de carpetas y archivos son las que se manipularan los datos de cada tabla que este en la base de datos seleccionada.

Cada archivo tiene una relación según la arquitectura del MVC la cual es la que se aplica para poder separar cada archivo de acuerdo a su función.

Que función tiene cada componente.

Los modelos poseen la estructura básica de las clases y estos solo se encargaran de encapsular los atributos de cada clase para una manipulación más segura y apegada a la programación orientada a objetos.

Su estructura de clase es la siguiente:

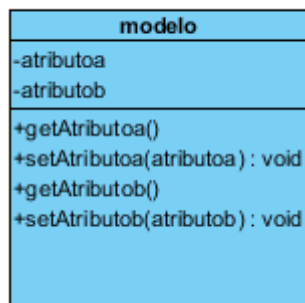
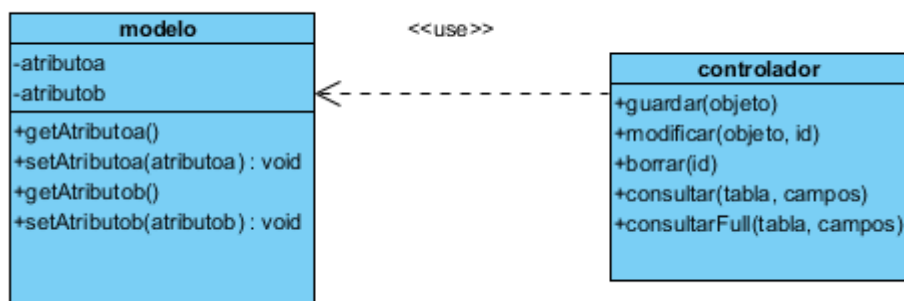


Fig. 10. Estructura de una clase

Los controladores se encargan de las funciones de la lógica del negocio, heredan atributos de los modelos porque así podrán manipular de una forma más conveniente los datos, su función principal es la de aplicar las funciones de CRUD sobre la base de datos de acuerdo al comportamiento del o de los modelos asociados.

Esta es su estructura según el modelo de clase:

Fig. 11. Modelo de clase.



La carpeta visto aloja una carpeta con el nombre de cada tabla en la cual se alojan 4 archivos de los cuales se utilizan para controlar la petición de datos, muestra de resultados y control de las funciones o acciones que el usuario ejecuta.

3. Probar la aplicación generada:

Al haber creado la estructura de carpetas además se crea un archivo denominado menu.php el cual aloja los vínculos a cada una de las páginas para realizar el mantenimiento de datos de cada tabla.

La siguiente imagen muestra el acceso a la página inicial de una las carpetas de vistas creadas con el framework.

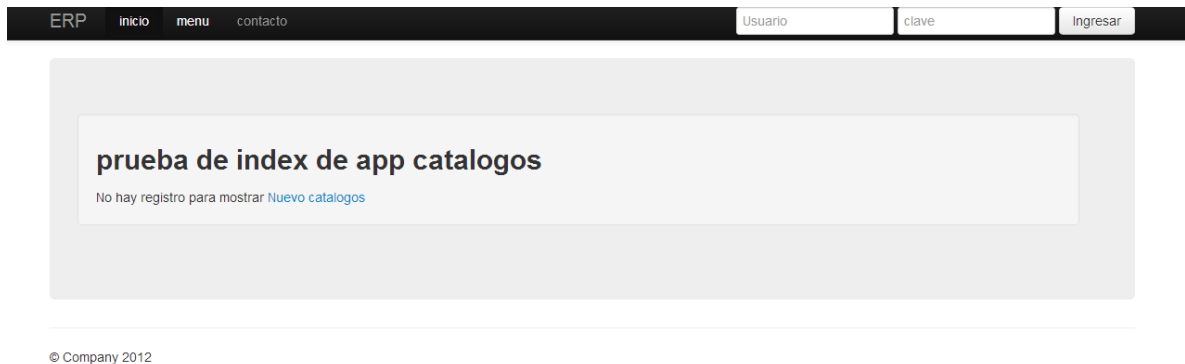
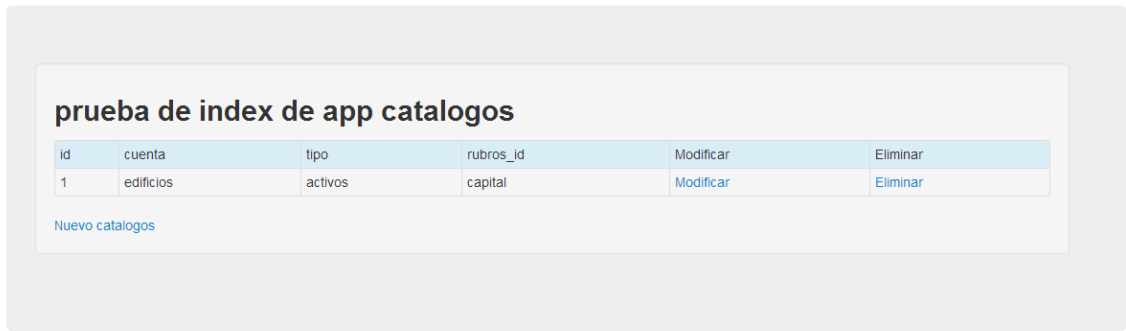


Fig. 12. Pantalla de menú de la aplicación generada

Como se muestra esta es la página principal para el mantenimiento de datos del componente seleccionado en este caso sería la tabla catálogos que se ha usado desde la base de datos con la que se creó la aplicación.

Al dar clic al vínculo podemos agregar un nuevo registro el cual al ser almacenado, regresara a esta página, la cual tendrá un cambio el cual será mostrar los datos que están almacenados en la tabla.



© Company 2012

Fig.13. Pantalla de mantenimiento de una tabla de la aplicación generada

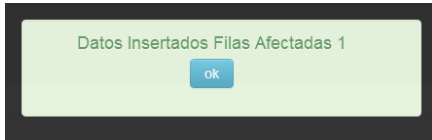
En este caso cuando hay datos ya podemos realizar otras acciones como, modificar dicho registro o eliminarlo según el caso.

Cuando se agrega un nuevo registro se muestra un formulario para capturar los datos, dicho formulario se genera de acuerdo al número de campos en la tabla a su vez dependiendo del tipo de campo genera un control para capturar de una forma más conveniente dicho datos, como se muestra en la imagen este es un formulario auto generado además posee automáticamente un elemento de validación para evitar almacenar datos inconsistentes.

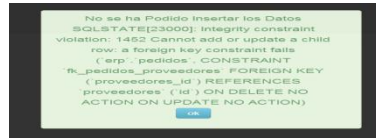
Formulario para captura de los datos.

Mensajes de validación de los datos

Al momento de almacenar un dato tenemos el siguiente mensaje el cual nos confirma el guardado o errores que pueden suceder a la hora de almacenar



Mensaje de confirmación



Mensaje al momento de ocurrir un error al realizar la acción en la base de datos

Controles generados para la aplicación:

Como se menciona antes los controles se generan de acuerdo al tipo de datos del campo de la base de datos en los casos los más comunes que encontraremos son los siguientes.

Casillas de texto: estas serán para datos comunes como textos o números	
Selectores de fecha: estos se generan cuando el dato es de tipo date, integran un calendario para poder buscar una fecha de una forma más cómoda, pero también se puede escribir la fecha en la caja de texto al cual está asociado el dato.	
Listas: este control se genera cuando en la base de datos se utiliza un tipo de datos denominado ENUM y este caso es para datos que se repiten y son fijos dentro de la información que se solicita	
Lista datos: este es un control el cual permite ver los datos que poseen relaciones con otras tablas no se generan por un tipo de dato sino por encontrar una llave foránea dentro de las especificaciones del campo en la tabla.	

Tabla. 5. Controles generados

Las funciones de modificar y eliminar se manejan desde la página de inicio de cada opción, poseen los mensajes anteriormente descritos y validan que los datos solicitados.

5.11 RECOMENDACIONES PARA MODIFICAR EL CÓDIGO DE LA APLICACIÓN GENERADA:

Para poder modificar los scripts generados o crear un módulo propio que se adapte a diferentes necesidades se recomienda.

- PHP y sus características orientadas a objetos.
- Si modifica las rutas de los archivos y carpetas, debe de adaptar las direcciones de importación en los elementos INCLUDE de cada página o script donde sea necesario.
- Si se modifican los archivos creados en vistas se recomienda mantener la lógica que genera el Framework, colocando los comportamientos, vistas o consultas separados.
- Puede generar sus códigos sin necesidad de usar el asistente solo debe de considerar usar los nombres definidos para las clases que debe importar además de las funciones por defecto que se deben adaptar para el manejo del CRUD.
- Para el uso de los controles se debe de saber que parámetros son básicos, y cuales opcionales ya que el uso de parámetros de forma errónea puede causar comportamientos inadecuados y/o errores de ejecución.

6. CONCLUSIONES.

Al concluir el desarrollo del proyecto, se puede considerar que:

- Se desarrolló, la primera versión del Framework, la cual permite y aumenta la factibilidad para desarrollo, de aplicaciones para registros y controles administrativos a empresas utilizando como herramienta de desarrollo el lenguaje PHP que aplican bases de datos.
- Se aplicó y analizó las tecnologías actuales para desarrollo de aplicaciones considerando como que se permita producir software según estándares actuales.
- Se aplicó diseño orientado a objetos y modelo vista controlador para el desarrollo del Framework
- Se realizó una documentación inicial para la publicación del software a la comunidad.

7. RECOMENDACIONES.

Se considera necesario:

- Publicar y poner a prueba el software creado, para su respectiva validación y desarrollo en al menos una comunidad de programadores.
- Se requiere completar la estandarización del código según la W3C de los componentes de visualización de las aplicaciones que genera el Framework.
- Promover la creación de este tipo de proyectos que motivan a generar aplicaciones propias en la industria del software que en El Salvador donde aún es insipiente.

8. REFERENCIAS:

8.1 BIBLIOGRÁFICAS:

Kenneth E. Kendall, Julie E. Kendall, Antonio Núñez Ramos. Análisis y Diseño de Sistemas. Sexta edición. 2005.

- Bahit Eugenia, POO y MVC en PHP, URL: <http://eugeniabahit.blogspot.com>

8.2 WEB:

- Estándares de diseño: <http://www.w3.org/standards/webdesign/htmlcss>. Consultado Julio 2012:
- Características de jquery : <http://es.wikipedia.org/wiki/JQuery#Caracter.C3.ADsticas>. Consultado Julio 2012:
- Plugin: <http://bassistance.de/jquery-plugins/jquery-plugin-validation/> Consultado Julio 2012:
- MVC: <http://blog.suenyos.com/2011/07/20/crear-un-framework-en-php5-desde-cero-1%C2%AA-parte/>
- Netbeans: <http://netbeans.org/> Consultado Julio 2012:
- Diseño de BD: <http://www.mysql.com/products/workbench/> Consultado Julio 2012:
- *Apache License Version 2.0, enero 2004, Fuente: http://www.apache.org/licenses/*

9. ANEXOS.

9.1 LICENCIA: APACHE LICENSE

Apache License

Version 2.0, January 2004

Fuente: <http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Este es el extracto que se colocara en el software.

Copyright [2012] [name of copyright owner]

*Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.*

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

9.2 CRONOGRAMA DE ACTIVIDADES

ACTIVIDADES	Plan (meses: junio/2012 a enero de 2013)								
	M	+1	+2	+3	+4	+5	+6	+7	+8...
• Conformación del equipo de trabajo		X							
• Actualización en nuevas tecnologías		X	X						
• Planificación del framework			X						
• Gestión de recursos y organización del equipo de trabajo.			X	X					
• Diseño del framework				X	X				
• Desarrollo, prueba y documentación del Framework					X	X	X		
• Publicación y validación del Framework									X
• Mantenimiento y evaluación del proyecto									X

www.itca.edu.sv



UN FUTURO LLENO DE OPORTUNIDADES

Escuela Especializada
en Ingeniería

ITCA  **FEPADE**

SANTA TECLA - ZACATECOLUCA - SAN MIGUEL - SANTA ANA - LA UNIÓN



www.itca.edu.sv

Sede Central Santa Tecla

Km. 11 Carretera a Santa Tecla.

Tel. (503) 2132-7400

Fax. (503) 2132-7599

MEGATEC La Unión

C. Santa María, Col. Belén, atrás del
Instituto Nacional de La Unión.

Tel. (503) 2668-4700

MEGATEC Zacatecoluca

Km. 64 1/2, desvío Hacienda El Nilo,
sobre autopista a Zacatecoluca y
Usulután. Tel. (503) 2334-0763, (503)
2334-0768 Fax. (503) 2334-0462

Centro Regional San Miguel

Km. 140, Carretera a Santa Rosa de Lima.

Tel. (503) 2669-2292, (503) 2669-2299

Fax. (503) 2669-0961

Centro Regional Santa Ana

Final 10a. Av. Sur, Finca Procavia
Tel. (503) 2440-4348, (503) 2440-2007
Tel. Fax. (503) 2440-3183