



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

---

---

**FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN**

**“DESARROLLANDO ADMINISTRADORES WEB”**

**DESARROLLO DE UN CASO PRÁCTICO**

**QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO EN COMPUTACION**

**PRESENTA:**

**PABLO ALBERTO RODRIGUEZ GUERRERO**

**ASESOR:**

**MAT. LUIS RAMIREZ FLORES**

**SAN JUAN DE ARAGÓN, EDO. MÉX., SEPTIEMBRE DE 2010**





Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## Agradecimiento

*"A todos y a cada una de las personas que colocaron su granito de arena para la realización de este logro, este éxito es compartido con cada uno de mis seres queridos que creen en mi y que ahora les rindo tributo por medio de esta victoria"*

## Índice

- **Introducción** .....1
- **Planteamiento del problema**.....3

### Capítulo 1 – Framework

1.1. ¿Qué es un Framework?.....	7
1.2. Clasificación.....	7
1.3. Desarrollo de un Framework.....	9
1.4. Web Framework.....	9
1.5. Una vez y sólo una.....	12
1.6. Patrón de Diseño.....	13
1.7. Ventajas al usar un Framework.....	13
1.8. Desventajas al usar un Framework.....	14
1.9. Panorama Global.....	14

### Capítulo 2 – Frontend

2.1. Problemas de compatibilidad.....	18
2.2. Usabilidad.....	19
2.3. Beneficios de la Usabilidad.....	20
2.4. Componentes de un Front End.....	20
2.4.1. HTML.....	20
2.4.2. JavaScript.....	21
2.4.3. AJAX.....	22
2.5. Cloud Computing.....	23
2.6. CSS.....	25
2.6.1. Ventajas al utilizar CSS.....	26

**Capítulo III – Backend**

3.1.CRUD.....28

3.2.Niveles de Usuario.....28

3.3.Wordpress.....29

3.4.Base de Datos.....30

    3.4.1.Ventajas.....30

3.5.MySQL.....31

    3.5.1.SQL.....33

**Capítulo 4 – Wizard**

4.1.Metaprogramación.....36

4.2.Scaffolding.....37

4.3.Generación de Código.....37

4.4.Dominio del Problema.....38

    4.4.1.Análisis.....39

    4.4.2.Requerimientos técnicos.....39

    4.4.3.Solución.....40

4.5.Demostración del Proyecto.....45

• **Conclusiones.....47**

• **Glosario.....48**

• **Bibliografía.....50**

## Introducción

Internet hoy en día es una herramienta que juega diferentes roles tanto en la sociedad, en la vida personal de cada individuo como en el sector privado y público de nuestro país y del resto del mundo, donde es inimaginable visualizar un presente ó un futuro donde esta herramienta no exista. Cada día aumenta el número de usuarios de esta red de redes, ya que en ella podemos encontrar una infinidad de herramientas que nos pueden ayudar con nuestras tareas del día a día.

Nos podemos apoyar en la Web<sup>1</sup> para realizar nuestras actividades rutinarias como bien puede ser comprar, pagar las tarjetas de crédito, el teléfono, la luz, mantenerse en contacto, guardar archivos, ver televisión en vivo, consultar el mapa de la localidad donde nos encontramos, jugar online, escuchar radio, descargar podcast, vigilar que nuestro hogar/negocio este seguro, y podría seguir listando una infinidad de tareas que podríamos realizar desde la comodidad de nuestro computador gracias a la Internet.

Nos podemos percatar que la tecnología Web a crecido a pasos agigantados la ultima década (2000 – 2010). Hoy en día hay una gran cantidad de empresas que realizan sus transacciones por la red, beneficiándose en el ahorro de mermas y logrando ser más competitivos.

Para lograr esto, las empresas se han visto en la necesidad de actualizarse, por lo general han migrado sus aplicaciones de una plataforma de escritorio a una plataforma Web, en donde es mucho mas practico el acceso remoto para la colaboración grupal con algún fin específico, obteniendo una mayor productividad ya que los empleados/usuarios pueden estar realizando sus labores desde un lugar distinto a las oficinas del corporativo.

Cada una de estas empresas cuentan con una área de desarrollo Web, la cual se encarga de fabricar el software que la empresa requiera para enfrentar grandes retos como lo es la seguridad de las aplicaciones y la compatibilidad entre navegadores, por mencionar algunas.

Una forma saludable de enfrentar esos retos es contar con piezas prefabricadas listas para aplicarse de tal forma que no se tenga que iniciar el desarrollo de una aplicación desde cero cada ocasión.

En la red de redes existe una gran variedad de piezas prefabricadas, pero en ocasiones la curva de aprendizaje resulta muy larga, por lo que tal vez sería mejor fabricar esa misma pieza con nuestras propias manos.

Sin embargo, al reutilizar las piezas prefabricadas se obtienen muchos beneficios, entre los que destacan el ahorro de tiempo, estandarización del código y por tanto una calidad superior, aumentando el tiempo de vida útil del producto.

Pero surge una directriz importante; “el tiempo es oro”, por lo que cada vez se requiere fabricar más, con mayor calidad, en menor tiempo y con el menor costo. Entonces ¿será posible fabricar software aun más rápido, garantizando la calidad del producto?

Esto se parece a un concepto de industrialización; incrementando las ventas, abaratando la mano de obra y reduciendo los precios al consumidor, con la finalidad de ser mas competitivos.

Lo que llevo al éxito la Revolución Industrial en el siglo XVIII fue la mecanización de industrias. Donde eran las máquinas las que realizaban y/o facilitaban el trabajo a los obreros, produciendo mucho más en menor tiempo.

---

<sup>1</sup> Web: red, en este caso sinónimo de Internet

Podemos percibir esta evolución donde primero era el hombre quien fabricaba directamente pieza por pieza, y después el hombre fabrica la máquina que se encargaría de fabricar las piezas de forma mecánica.

A lo largo de este trabajo se explicará como se llega a la “industrialización del software”. Por supuesto que hablamos de la fabricación de una aplicación para la automatización de la creación de otra, este concepto es conocido como Metaprogramación.

Esta aplicación que fabrica otra, se le conoce como Wizard. El cual nos guía a través de pasos asistidos para la fabricación de software, sin la necesidad de involucrarnos con los detalles del producto. Por lo que será mucho más sencilla la línea de producción al dar por hecho el buen funcionamiento de las piezas prefabricadas.

De una manera algo sosa, se puede visualizar a un Wizard como un molde, tu solo necesitas agregar el contenido en el recipiente y el se encargara de darle forma, una y otra vez, cuantas veces sea necesario. Asegurando que todos los productos terminaran teniendo la misma estructura.

Mencionaba la gran variedad y cantidad de aplicaciones que existen en la Internet, cada una de estas tiene algún propósito en particular para la cual fue creada. En base a mi experiencia profesional existen cuatro operaciones básicas por lo general en cada sistema; estas son altas, bajas, cambios y borrado de información.

Invito al lector a descubrir a través del desarrollo de este trabajo, como es posible la fabricación de un Wizard, que se encargue de facilitar la producción de Software.

## Planteamiento del problema

Actualmente el uso de las mejores practicas para las empresas tanto para profesionistas es un tema primordial, dentro del desarrollo de Software es algo que se debe de tener muy presente, y dejar detrás la “codificación espagueti”, que en algún momento haría que se autodestruya la aplicación, impidiendo que alguien más que no haya participado en su desarrollo inicial le pueda dar un correcto mantenimiento, y mucho menos tener la posibilidad de escalar o migrar de plataforma dicha aplicación.

La generación de “código sucio” no es redituable para nadie, ya que estamos diciendo que se tendría que codificar prácticamente desde cero cada aplicación que se requiera, lo cual es un gran problema porque se traduce en horas hombre y por lo tanto en un costo extra para la empresa.

Todo desarrollo de software debería de pasar por un proceso de calidad pero en la práctica no siempre es así, ya que no solo depende del programador sino también de los líderes de proyecto y del gerente del área de desarrollo, quienes muchas veces carecen de una metodología de trabajo.

Hablando específicamente del desarrollo de aplicaciones para el Internet es verdaderamente un reto cumplir con una serie de requerimientos mínimos básicos para garantizar un correcto funcionamiento del producto en los distintos navegadores Web del mercado o por lo menos en los más populares venciendo los problemas de compatibilidad.

Estamos diciendo que escribir código sin basarse en una metodología y sin un análisis profundo de los requerimientos desencadena una serie de problemáticas, las cuales se traducen en mermas de tiempo, mayores costos, incumplimiento de contratos, estrés laboral, además de un producto obsoleto en poco tiempo, entre muchas otras cualidades negativas.

Por desgracia la generación de “código espagueti” es muy común en la industria debido al poco entrenamiento con las herramientas de desarrollo, desde el lenguaje de programación, el IDE<sup>1</sup>, hasta la metodología de desarrollo y el Framework<sup>2</sup>.

El no tener definidas las herramientas de trabajo óptimas y carecer de una metodología, puede resultar en un análisis corto en comparación con los objetivos que se desean alcanzar. Sí el análisis, que es una fase primordial en la fabricación de Software y en la que más tiempo se debe de invertir, llega a resultar errónea se fijarán lapsos inadecuados para el desarrollo del producto, orillando a liberar el proyecto de acuerdo a las posibilidades y capacidades, cayendo muy probablemente en la redundancia de código, lo cual simboliza un producto falto de calidad.

La redundancia de código dentro de una misma aplicación y dentro de una serie de aplicaciones relacionadas, es algo que se debe de evitar a toda costa, esto hará mucho más fácil el mantenimiento del producto abriendo las posibilidades de potenciar la aplicación en un futuro.

La redundancia puede ser provocada por las malas prácticas, por no parametrizar ó al no manejar módulos centralizados y reutilizables en la misma aplicación e incluso en otras posteriores.

Cuando se codifica una aplicación sin parametrizar queda casi inentendible para alguien más, a lo que comúnmente se le conoce como “código sucio” y no reutilizable, con lo que automáticamente está incrementando los costos de desarrollo, ya que cada línea de código generada y cada hora invertida será desechada a la basura en el momento de requerir escalar la aplicación.

---

<sup>1</sup> IDE: entorno de desarrollo de Software

<sup>2</sup> Framework: código prefabricado, considerado una herramienta de trabajo.

En caso de que ese mismo programador sea transferido a otro proyecto o que decida renunciar a la empresa, será todo un reto para su reemplazo llevar a cabo una tarea de mantenimiento de la aplicación y por ende la detección de bugs<sup>3</sup>, complicando demasiado la tarea de debug<sup>4</sup>.

Este problema puede incluso extenderse afectando a todo el equipo de desarrollo y entorpecer un trabajo de colaboración, retrasando la entrega de un mismo proyecto o de otros. De forma particular el mal rendimiento de un software provoca que esto sobrepase los límites de la empresa fabricante, afectando de manera directa al consumidor, aumentando la curva de aprendizaje del usuario final generando una mala imagen tanto del producto como del fabricante.

Un ejemplo del desprestigio de la aplicación como del fabricante es lo que ha pasado durante décadas con el sistema operativo de Microsoft Windows, en donde el fabricante se ha visto obligado a liberar una nueva versión de software con las mejoras necesarias para paliar las carencias de la versión anterior, lo cual no siempre es la solución, ya que después del rotundo fracaso del sistema operativo Windows Vista, se liberó lo antes posible Windows 7, el cual tampoco funcionó como se esperaba lo que ha evidenciado que solo ha sido una salida rápida para las exigencias de sus millones de usuarios.

Retomando el tema del desarrollo de Software para Web, se pueden generar problemas muy particulares, como por ejemplo que el sitio Web no sea indexable<sup>5</sup> por los buscadores como Google, Yahoo y Bing. Otro error que se genera a menudo es una incompatibilidad de la aplicación con los diferentes navegadores del mercado. Además se pueden generar problemas más graves como lo son los hoyos de seguridad al tener un mal manejo de sesiones y/o de conexión a la base de datos; orillando a quien se dedica a esta profesión a enfrentar el reto de dar soluciones de software en el menor tiempo, debido a que las consecuencias se pueden intensificar dependiendo del número de usuarios en línea.

En caso de que no se cuente con un buen kit de herramientas(Framework) que ayude a resolver los problemas seguridad o simplemente por problemas de compatibilidad derivados por las actualizaciones de los Navegadores y/o tecnologías Web, entonces se convertirá en una tarea que requerirá esfuerzos fuera de lo ordinario.

Al paso del tiempo el programador encontrará en las aplicaciones que muchos módulos son muy parecidos y que realmente su lógica resuelve la misma acción con distintos datos, aun trabajando con una metodología Orientada a Objetos se percatara de esta redundancia en el código.

Incluso si se recurre a la Herencia<sup>6</sup>, se verá que no se podrá evitar la existencia de código semejante. Se llegará a reutilizar algún código existente y se gastará tiempo modificando una "Clase"<sup>7</sup> la cual era muy similar a la que se necesitaba pero que no estaba diseñada para el dominio del problema en específico.

Un ejemplo claro es cuando se fabrica un panel administrador (Backend<sup>8</sup>), ya que se encontrará con un repetido desarrollo de código para tareas semejantes. Se programará un módulo para insertar información a la Base de Datos, otro para seleccionar, uno más para actualizar y por último uno que permita eliminar. En el mejor de los casos cada uno de estos módulos se repite por cada una de las tablas que componen a la Base de Datos, pero podría existir una redundancia de código dependiendo de los requerimientos de la aplicación.

---

<sup>3</sup> bugs: errores de software no detectados en tiempo de fabricación y que son encontrados por lo regular cuando la aplicación se encuentra en producción.

<sup>4</sup> debug: termino utilizado en el desarrollo de Software para referirse al proceso de búsqueda de errores en alguna aplicación.

<sup>5</sup> Termino comúnmente usado por los buscadores Web, para decir que la información publicada dentro de un sitio ha sido encontrada, para una posterior entrega de un resultado rápido y eficiente, de esta manera la información se hace visible para el usuario final por medio de los buscadores.

<sup>6</sup> Termino que se refiere a la reutilización de código existente con la posibilidad de extenderlo para que cumpla con nuevos requerimientos.

<sup>7</sup> Termino usado en el desarrollo de Software para referirse a una colección de código con un propósito específico.

<sup>8</sup> Concepto usado para denominar al panel administrador de un sitio Web, y de manera intrínseca su definición incluye a la base de datos.

El desarrollo de este proyecto tiene la finalidad de abatir muchos de los problemas generados por las malas prácticas del desarrollo de Software. En específico, es un proyecto que automatiza la generación de código estándar, por medio de un Wizard<sup>9</sup>, que nos guía por medio de una interfaz gráfica, obteniendo como resultado un código uniforme listo para ponerlo en producción y realizar las tareas de lectura, escritura, edición y borrado de información de una base de datos, a estas cuatro operaciones en ingles se le conoce como CRUD.

El Wizard será el encargado de realizar toda la labor, para que el programador no tenga la necesidad de escribir ni una sola línea de código para generar un panel administrador, lo cual ahorra mucho tiempo en la fase de pruebas y de análisis al igual que tiempo de desarrollo, todo eso es posible a partir del uso y dominio de un Framework, que nos permita la generación de un Backend y un Frontend usable, además de permitir la compatibilidad de la aplicación entre navegadores Web.

El tema central de este proyecto es la generación automatizada de código para las tareas más comunes de realizar dentro de un Backend, refiriéndonos específicamente al CRUD, estando fuera del alcance de este proyecto el desarrollo y/o demostración de cualquier otra característica. Sin embargo será necesario abarcar un marco teórico que permita esclarecer la utilización de ciertas herramientas y tecnologías además de justificar la selección de las mismas.

Tanto este documento como el proyecto esta dirigido a los programadores Web, con la finalidad de que por medio del presente cuenten con las armas necesarias para hacer frente a la vida profesional dentro del ramo de Desarrollo de Software con herramientas como PHP y MySQL.

Es complicado satisfacer a un cliente que no sabe lo que quiere y mucho menos lo que realmente necesita, pero que te exige lo mejor. Es por eso que es de primera necesidad contar con una buena caja de herramientas.

Durante mi ejercicio profesional he tenido la responsabilidad de mantener, modificar y extender la funcionalidad de un Software fabricado por otra persona que no conozco, por lo que, aun teniendo una metodología de desarrollo, será complicado de primera instancia entender su lógica, ya que programar se basa en como concebimos una solución a un problema y todos razonamos la información de formas distintas.

Esencialmente me percate de que no es necesario reescribir lo que ya hemos programado, sí dentro de un programa, ya tienes una herramienta que realiza cierta tarea, definitivamente puedes reutilizarla en otro. A esto se le llama reutilización de código, pero una forma peculiar en la practica es por medio de “un programa que genere el código que ejecutara otro programa” (Metaprogramación).

Dicho de otra manera es un concepto usado en la película de Matrix, donde las máquinas que fueron fabricadas por el hombre, después producen más máquinas totalmente independientes del ser humano.

Al programa que genera programas se le conoce como Wizard, cabe mencionar que solo fabrica programas de características bien definidas y limitadas, pero suficiente para tener el beneficio de ahorrar varias horas laborales, además de obtener código uniforme, estándar, fácil de mantener, modificar y de extender su funcionalidad.

A lo largo y ancho de este documento explicaremos como fabricar un Wizard que nos ayude con las tareas de altas, bajas, cambios y borrado de la información dentro de un panel administrador. Por lo que no tendremos que empezar desde cero y la próxima vez que nos asignen el desarrollo de un panel administrativo ya tendremos aventajado el proyecto en un porcentaje bastante considerable.

---

<sup>9</sup> Wizard es una aplicación que se encarga de fabricar a otra, dentro de un dominio de problema específico.

# Framework



*"una Clase es a un Objeto, como un Framework es a una Aplicación"*<sup>1</sup>

---

<sup>1</sup> <http://www.jonathancrossland.com/page/what-is-a-framework.aspx>

## ¿Qué es un Framework?

Vamos a hacer una analogía de lo que es un Framework, por ejemplo un arquitecto cuenta con sus diferentes tipos de estilógrafos, uno o varios restiradores, juegos de escuadras, compás, Software especializado para el diseño y simulación, entre muchas otras herramientas con las que se apoya para realizar su trabajo. Nos queda claro que cada una de estas herramientas tiene un propósito específico, pero que todas conviven en conjunto, manipuladas por el arquitecto resuelven problemas de un dominio específico.

Un arquitecto con sus herramientas no podría reparar la tubería de agua de un edificio sin embargo si podría proporcionarle los planos al plomero indicándole con exactitud la ubicación de la tubería a lo largo y ancho de las instalaciones, facilitando la ubicación de la posible falla.

En nuestro ejemplo podemos entender que un Framework es una caja de herramientas las cuales nos ayudan a resolver una serie de problemas relacionados.

En el ámbito del desarrollo de software un Framework es una colección de clases. Donde una clase equivale a una herramienta, que permite la ágil construcción de aplicaciones, es diseñado con el intento de facilitar el desarrollo de software, permitiendo a los programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel para proveer un sistema funcional.

Por ejemplo, un equipo de desarrollo de software que usa un Framework para desarrollar un sitio Web, puede enfocarse en cómo las reglas de negocio van a funcionar en lugar de preocuparse por una interfaz gráfica de usuario libre de errores.

Vamos hacer un paréntesis para explicar lo que en desarrollo de software se le conoce como “clase”.

```

1  <?php
2
3  class Calculadora{
4
5      var $a; /*operando a*/
6      var $b; /*operando b*/
7      var $r; /*resultado*/
8
9      function suma($a,$b);
10     function resta($a,$b);
11     function multiplicacion($a,$b);
12     function division($a,$b);
13 }
14
15 ?>
```

Fig I.1 Clase Calculadora

En la Fig. I.1 vemos el código fuente de la clase (class) Calculadora, donde se definen dos variables a y b que son los operandos. Una tercer variable que es donde se almacenara el resultado de la operación realizada. Además se definen cuatro funciones (function) suma, resta, multiplicación y división.

La clase Calculadora (Fig. 1.1) actúa como una plantilla ya que puede ser reutilizada en más de una aplicación, por lo que en el desarrollo de nuestra siguiente aplicación podremos enfocar nuestros esfuerzos en los nuevas características que sean solicitadas, sin preocuparnos por lo que ya esta hecho y probado.

Un Framework es una colección de plantillas que se relaciona directamente con un dominio específico, es decir, con una familia de problemas relacionados. Como cualquier otra plantilla tienen la capacidad de ser reutilizados una y otra vez, sirviendo como plataforma, ahorrando tiempo. A esto se lo conoce como reutilización del *código fuente* (*source code*).

En otras palabras, un Framework es un conjunto de bloques de software prefabricados comúnmente nombrados "librerías" que los programadores pueden utilizar para ampliar o personalizar soluciones informáticas específicas. Con un Framework los desarrolladores no tienen que empezar desde cero cada vez que se escribe una aplicación, ya que la colección de clases definidas dentro del Framework pueden ser reutilizadas, tantas veces sea necesario para una o varias aplicaciones.

Al reutilizar un Framework, tienes la ventaja y la garantía de que se está respetando cierto estándar lo cual se traduce como calidad de Software y a su vez como ahorro de tiempo para los programadores, disminuyendo los costos de producción generando mayores ganancias.

Supongamos que hay 100 aplicaciones escritas bajo el mismo Framework, y que hay un error provocado por un archivo del Framework, entonces estamos hablando de un error multiplicado 100 veces, el cual sería muy sencillo reparar con tan solo editar un solo archivo. Lo cual no sería tan fácil si tuviéramos que editar 100 archivos independientes.

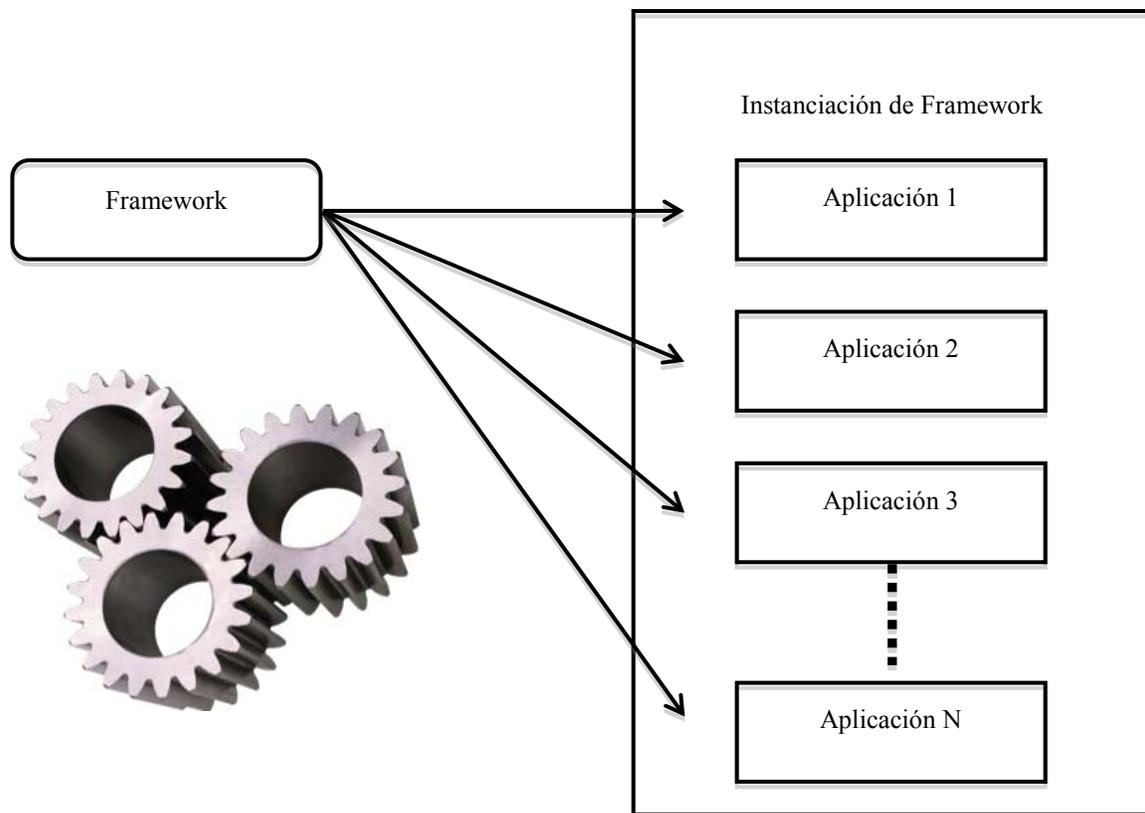


Fig. I.2 Ilustra la reutilización de código a partir de un Framework

## Clasificación

Un Framework se puede clasificar según su extensibilidad; puede ser utilizado como una caja blanca o caja negra.

- El Framework de caja blanca también llamado Framework con arquitectura de configuración conducida, donde uno puede mandar a llamar las clases definidas en el Framework a través de la herencia o composición. Por lo que el programador debe enfocarse en aplicar las herramientas que te proporciona el Framework, y debe de entenderlo muy bien para poder desarrollar una aplicación.
- El Framework de caja negra es aquel donde una herramienta automática crea las clases y el código fuente. Por ejemplo, es posible utilizar un Wizard o configurador gráfico que dirija al usuario gradualmente paso por paso. La caja negra del Framework no requiere que el usuario sepa los detalles internos del Framework. Por lo tanto, este tipo de Framework también se llama Framework de datos conducidos y son generalmente más fáciles de usar.
- El Framework que contiene características de ambas cajas se llama Framework de caja gris.

Algunos Frameworks que no son *open source* tienen la característica de que no son mutables ni tampoco pueden ser alterados fácilmente. Estos puntos inmutables constituyen el núcleo o kernel de un Framework, también llamados como los puntos congelados o frozen-spots del Framework.<sup>2</sup>

## Desarrollo de un Framework

Es importante tener presente que el desarrollo de un Framework será por lo menos tan costoso como el solo desarrollo de la aplicación. Sin embargo el esfuerzo de construir un Framework puede tener éxito a través de la construcción repetida de aplicaciones dentro del mismo dominio de problema.

Cuando se opta por el desarrollo de un Framework propio, uno debe preguntarse:

¿Se crearán aplicaciones de este mismo dominio más de una vez?

Si la respuesta es si, entonces muy probablemente el desarrollo de un Framework sea la solución. Sin embargo, habrá que analizar los costos contra las ventajas de elegir desarrollar un Framework propio en vez de desarrollar software hecho a la medida.

Además se tendrá que analizar la posibilidad de usar un Framework existente para nuestro dominio de problema, y tratar de ahorrar el mayor tiempo posible calculando el tiempo de desarrollo de un Framework propio contra la curva de aprendizaje de un Framework existente.

## Web Framework

Un Framework para aplicaciones Web es una librería de software diseñada para apoyar al programador con el desarrollo de sitios Web dinámicos. Tiene la finalidad de apoyar en las actividades comunes usadas en los desarrollos. Como lo es la gestión de sesión de usuario, el almacenamiento de datos y un sistema de plantillas.

---

<sup>2</sup> El Desarrollo del Framework Orientado al Objeto <http://www.acm.org/crossroads/espanol/xrds7-4/frameworks.html>

Como habíamos mencionado anteriormente un Framework esta directamente vinculado a facilitar las herramientas para solucionar una familia de problemas relacionados. De igual manera y en la mayoría de los casos esta ligado con un solo lenguaje de programación. Algunos de los lenguajes de programación mas populares en la actualidad para el desarrollo de páginas Web dinámicas son:

 PHP Es un lenguaje de scripting de propósito general ampliamente utilizado. Está diseñado especialmente para desarrollo Web y puede ser incrustado dentro de código HTML. Es un proyecto open source por lo que se puede descargar sin costo alguno. La versión más reciente de PHP es la 5.3.1 al 19 de Noviembre de 2009.<sup>3</sup>

Ejemplo de Código **PHP**:

```
<? echo "Hola Mundo usando PHP"; ?>
```



Java<sup>4</sup> es un lenguaje de programación orientado a objetos desarrollado por pequeño grupo de ingenieros denominado “Green Team” por Sun Microsystems a principios de los años 90.<sup>5</sup>

Este lenguaje de programación tiene un slogan bastante llamativo para cualquier desarrollador. “Write once, run anywhere” (WORA). Que implica que nos da las herramientas para que los programadores desarrollen sus aplicaciones sin tener que preocuparse si van a correr en un sistema operativo Windows, Mac, Linux u otro.

Oficialmente Java de Sun Microsystems nace en 1995.<sup>6</sup>

Ejemplo de Código **Java**:

```
public class Hola {
    public static void main(String[] args){
        System.out.println("Hola Mundo usando Java");
    }
}
```

<sup>3</sup> Sitio Oficial de PHP <http://php.net/>

<sup>4</sup> Sitio Oficial de Java <http://java.sun.com/>

<sup>5</sup> Historia de Java <http://java.com/en/javahistory/index.jsp>

<sup>6</sup> Java en la línea del tiempo <http://java.com/en/javahistory/timeline.jsp>



C# es un lenguaje de programación orientado a objetos desarrollado por Microsoft como parte de su plataforma .NET.

Su sintaxis es similar al de Java aunque incluye mejoras derivadas de otros lenguajes.<sup>7</sup>

Ejemplo de Código **C#**:

```
class MainApp {
    public static void Main() {
        Console.WriteLine("Hola Mundo usando C#");
    }
}
```

Como lo acabamos de mencionar PHP, Java y C# son de los lenguajes de programación más populares para el desarrollo de aplicaciones basadas en una plataforma Web. Por lo tanto aquí tenemos una lista de los Frameworks más utilizados por los desarrolladores Web.

**PHP Framework**<sup>8</sup> 

Proyecto	Sitio Web Oficial
CakePHP	<a href="http://cakephp.org/">http://cakephp.org/</a>
PEAR	<a href="http://pear.php.net/">http://pear.php.net/</a>
Symfony	<a href="http://www.symfony-project.org/">http://www.symfony-project.org/</a>
Zend	<a href="http://www.zend.com/">http://www.zend.com/</a>
Zoop	<a href="http://zoopframework.com/">http://zoopframework.com/</a>
Prado	<a href="http://www.pradosoft.com/">http://www.pradosoft.com/</a>
Qcodo	<a href="http://www.qcodo.com/">http://www.qcodo.com/</a>
Akelos	<a href="http://www.akeos.org/">http://www.akeos.org/</a>
Yii	<a href="http://www.yiiframework.com/">http://www.yiiframework.com/</a>
Codeigniter	<a href="http://www.codeigniter.com/">http://www.codeigniter.com/</a>

<sup>7</sup> Sitio Oficial en Español de Microsoft .Net Framework

<sup>8</sup> Top 10 Ranking PHP Frameworks <http://www.phpframeworks.com/top-10-php-frameworks/>



Proyecto	Sitio Web Oficial
Apache Cocoon	<a href="http://cocoon.apache.org/">http://cocoon.apache.org/</a>
Apache Struts	<a href="http://struts.apache.org/">http://struts.apache.org/</a>
AppFuse	<a href="http://appfuse.org/display/APF/Home">http://appfuse.org/display/APF/Home</a>
Eclipse RAP	<a href="http://www.eclipse.org/rap/">http://www.eclipse.org/rap/</a>
JBoss Seam	<a href="http://seamframework.org/">http://seamframework.org/</a>
OpenLaszlo	<a href="http://www.openlaszlo.org/">http://www.openlaszlo.org/</a>
OpenXava	<a href="http://www.openxava.org/web/guest/home">http://www.openxava.org/web/guest/home</a>
Spring	<a href="http://www.springsource.org/">http://www.springsource.org/</a>
Grails	<a href="http://www.grails.org/">http://www.grails.org/</a>



.NET es una Tecnología propietaria de la compañía fundada por Bill Gates, Microsoft. Una de las ventajas que ofrece este Framework es la posibilidad de integración de otros lenguajes de programación como por ejemplo PHP, C++, ASP, Visual C++, entre otros.

### Una vez y sólo una

El principio DRY<sup>9</sup> (en inglés Don't Repeat Yourself ) es una filosofía que promueve la reducción de la duplicación. Según este principio ninguna pieza de información debería estar duplicada nunca, debido a que la duplicación incrementa la dificultad en el mantenimiento y migración posterior.

Cuando el principio DRY se aplica eficientemente y se mantiene la información de manera centralizada fomentando la reutilización de código resulta que cualquier cambio que se requiera en el sistema por lo regular tendrá que ser en un solo lugar.

Siempre será mas fácil editar un único archivo que editar un archivo por cada aplicación desarrollada.

<sup>9</sup> Dont Repeat Yourself <http://c2.com/cgi/wiki?DontRepeatYourself>

## Patrón de Diseño

Modelo Vista Controlador (MVC)<sup>10</sup> es un patrón de diseño de software introducido por la compañía Xerox PARC, posteriormente fue implementado en el lenguaje de programación Smalltalk.<sup>11</sup>

Tiene la finalidad de separar los datos de una aplicación de la interfaz de usuario y la lógica de control, convirtiéndolos en tres componentes distintos.

El patrón MVC es frecuentemente aplicado en el desarrollo Web, donde la “vista” es la página HTML y el código que provee de datos dinámicos a la página. El “modelo” es el Sistema Gestor de Base de Datos en conjunto con la lógica de negocio, y el “controlador” es el responsable de cachar los eventos de entrada desde la vista.<sup>12</sup>

## Ventajas al usar un Framework

1. Desarrollo ágil de aplicaciones. Las herramientas incluídas en un Framework fomentan la modularidad del código, eliminando las tareas repetitivas como lo son la conectividad de base de datos y el manejo de sesiones de usuarios, reduciendo de manera considerable el tiempo del desarrollo.

2. Reutilización de código fuente. Al tener un código modular será mucho más fácil hacer uso de un modulo de forma independiente cuando sea necesario para la aplicación sin necesidad de reescribirlo.

3. Facilidad de Mantenimiento. Al basar el desarrollo de una aplicación en un Framework, estamos garantizando que se están respetando políticas de fabricación de software. Lo que resulta en la generación de código uniforme fácil de modificar, probar, depurar y de mantener ya sea por el mismo desarrollador de la aplicación o por otro que no haya participado en el desarrollo inicial.

4. Portabilidad de aplicaciones de una arquitectura a otra. Por ejemplo, los bytecode generados a partir del código fuente de clases en Java pueden ser ejecutados sobre cualquier máquina virtual, independientemente de la arquitectura hardware y software subyacente.

5. Fiabilidad y Costos. El uso de código que ya ha sido construido, probado y utilizado por otros programadores aumenta la fiabilidad y reduce el tiempo de programación además de reducir los Bugs en el producto final. Para una empresa la reutilización de código se traduce en un menor costo.

6. Especialización del equipo de desarrollo. Los equipos de desarrollo de software se pueden dividir entre los que desarrollan el Framework y los que desarrollan la aplicación completa final. Esta separación de tareas permite que cada equipo se centre en objetivos más específicos y hacer uso de sus fortalezas individuales. Por ejemplo, los programadores que son expertos en el diseño de interfaz de usuario pueden trabajar en la usabilidad de la aplicación, mientras que las pruebas de los expertos de seguridad fortalecen el Framework en el que se basa la aplicación.<sup>13</sup>

7. Seguridad. un Framework puede proporcionar características de seguridad que se requieren a menudo de una clase común en las aplicaciones. Esto proporciona que todas las aplicaciones escritas con el Framework se beneficien de la seguridad sin el tiempo extra y el costo de su desarrollo. Un par de ejemplos comunes son la gestión de sesión de usuario segura y las conexiones a base de datos.<sup>14</sup>

---

<sup>10</sup> Model-View-Controller <http://ootips.org/mvc-pattern.html>

<sup>11</sup> MVC Xerox PARC <http://hubpages.com/hub/MVC>

<sup>12</sup> MVC <http://java.sun.com/blueprints/patterns/MVC-detailed.html>

<sup>13</sup> Framework – DocForge <http://www.docforge.com/wiki/Framework>

<sup>14</sup> ibídem.

8. Mejores Prácticas. Un Framework contribuye en la programación para diseñar los patrones, normas y en general las mejores prácticas. Un Framework de interfaz gráfica de usuario de escritorio, por ejemplo, puede crear automáticamente las barras de herramientas y botones comunes para el sistema operativo local. Un Framework de aplicación Web puede ayudar con cifrado de contraseñas de usuario o procesamientos de pagos.<sup>15</sup>

9. Mejoras. Las actualizaciones a un Framework pueden mejorar a la funcionalidad de la aplicación sin programación adicional por el desarrollador de la aplicación final. Si, por ejemplo, un Framework de comercio electrónico ofrece un nuevo método de pago, esta opción automáticamente puede estar disponible para el usuario final sin necesidad de programación adicional por el desarrollador de aplicaciones.<sup>16</sup>

10. Velocidad de desarrollo, facilidad de escalabilidad, calidad y estandarización de la aplicación final.

### **Desventajas al usar un Framework**

1. Dependencia de la aplicación, si se elige cambiar de Framework deberá reescribirse en su mayoría.

2. Demanda de grandes capacidades de procesamiento, debido a que un módulo genérico tiende a ser extenso, con la reutilización de código provoca un consumo de recursos “innecesarios”.

3. Degradación del rendimiento, al demandar grandes capacidades de procesamiento tal cual puede ocurrir con un módulo generalizado por no estar optimizado para un caso específico.

4. Curva de Aprendizaje, algunos Frameworks tienen una curva de aprendizaje muy alta.

5. Framework Inadecuado, algunos Frameworks son tan rígidos y altamente estructurados, que la elección inadecuada de ellos para un proyecto puede ser desastroso. Esto no es culpa del Framework, pero algunos son más adecuados en general y flexibles que otros.<sup>17</sup>

6. Bugs, los errores y problemas de seguridad en un Framework pueden afectar a todas las aplicaciones que lo utilizan. Por lo que cada error (Bug) en el Framework estará propagado N veces, por cada aplicación que haya hecho uso de él. Lo que puede ser muy riesgoso cuando se habla de seguridad.<sup>18</sup>

### **Panorama Global**

Hay que estar conciente de que los Frameworks tienen sus pros y sus contras, debido a que no son la solución para todos los problemas. Además, si está considerando usar una caja de herramientas ya existente en el mercado, estudie su documentación y verifique si hay una buena explicación (how to), si es que existe. También es importante observar las aplicaciones que fueron desarrolladas y el tiempo que les tomo este proceso. Esta información será fácil de conseguir si existe una amplia comunidad de usuarios, los cuales también nos pueden apoyar con nuestras dudas.

---

<sup>15</sup> ibídem.

<sup>16</sup> ídem.

<sup>17</sup> Framework – DocForge <http://www.docforge.com/wiki/Framework>

<sup>18</sup> ibídem.

# Frontend



Para el desarrollo de aplicaciones Web, hoy en día es de suma importancia contar con un patrón de diseño, que nos permita separar los datos y la lógica, de la interfaz gráfica de usuario. El patrón de diseño; Modelo Vista Controlador (MVC) es una solución para este problema, ya que basándonos en él, logramos tener de forma centralizada cada dato que compone nuestra aplicación. Por ejemplo, si necesitamos modificar algo relacionado con la interfaz gráfica de usuario, por lo general solo tendríamos que dirigirnos al archivo CSS y modificar el estilo que necesitamos, lo cual provocaría que todos los elementos HTML que estén usando ese estilo se vean afectados, lo cual genera un gran ahorro de tiempo a los programadores y diseñadores al hacer modificaciones al código. Lo que se traduce como un menor costo de mantenimiento de la aplicación y por tanto, mucho más rentable.

Dentro de este capítulo vamos a enfocarnos a la interfaz gráfica de usuario, que en el desarrollo de aplicaciones Web comúnmente se le conoce como Frontend, el cual se puede percibir como todos aquellos elementos que forman a una página Web, los cuales hacen posible una comunicación entre el usuario y el servidor Web, donde realmente radica la aplicación que se encarga de atender las peticiones del usuario, todo esto es posible por medio de un navegador Web, como lo son Internet Explorer, Firefox o Safari, entre muchos otros.

Para entender mejor el concepto de Frontend, primero vamos a ver algunos ejemplos, basándonos en algunos de los sitios Web más concurridos hoy en día.



fig. II.1 Frontend del sitio [www.google.com](http://www.google.com)

Google (*fig II.1*), el motor de búsquedas más usado por los internautas hoy en día, su Frontend se distingue por su simplicidad en su diseño que solo se compone esencialmente por un logotipo que identifica a la compañía, una caja de texto para colocar palabras claves de lo que estamos buscando, y un botón para ejecutar la búsqueda. Un Frontend muy simple, fácil de entender su uso y muy eficiente en sus resultados.



fig. II. 2 Frontend del sitio [www.twitter.com](http://www.twitter.com)

Twitter (*Fig II.2*) es un sitio de microblogging, el cual permite publicar textos muy cortos dentro de una red de seguidores, actualmente es considerado una red social. En este Frontend podemos denotar el uso de un

diseño Web 2.0, el cual se distingue por usar colores vivos y pastel además de usar textos con un tamaño de fuente fácil de leer. Es un Frontend amistoso con el usuario, ayuda a entender rápidamente donde se encuentra cada elemento dentro de la pagina Web, facilitando su uso.



fig II.3 Frontend del sitio [www.facebook.com](http://www.facebook.com)

Facebook (fig II.3) una de las redes sociales mas reconocidas a nivel mundial, cuenta con un Frontend con tonos de color que invitan al usuario a permanecer mas tiempo en el sitio ya que no lastiman la pupila del internauta, además cuenta con una serie de formularios HTML para recavar todo tipo de información a la cuenta del usuario para luego poderla compartir con las cuentas de su familia y amigos.

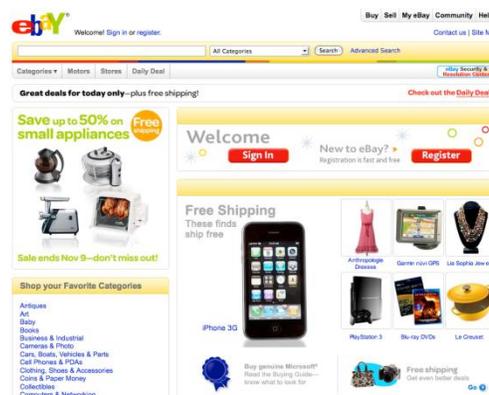


fig. II.4 Frontend del sitio [www.ebay.com](http://www.ebay.com)

eBay (fig. II.4) un sitio Web que nos permite tanto vender como comprar, productos y servicios. Se necesita tener una interfaz gráfica realmente intuitiva para que todo tipo de usuario pueda realizar sus transacciones además de inspirar confianza a los usuarios para hacer sus compras con tarjeta de crédito, sin duda es un Frontend muy bien trabajado.

De acuerdo a los ejemplos anteriores tomados de Internet, se entiende que un Frontend es la interfaz gráfica de usuario que permite a un Internauta desde un computador, por medio de elementos como botones, imágenes, cuadros de textos, links, entre otros, poder interactuar con un sistema que reside en un servidor Web. En otras palabras un Frontend es la parte visible del sitio Web para cualquier usuario que puede acceder por medio de los distintos navegadores que existen en el mercado, como por ejemplo Internet Explorer, Firefox, Safari, por mencionar algunos.

Cuando se fabrica software es muy importante seguir una metodología, la cual te va ahorrar varios dolores de cabeza, pero cuando hablamos específicamente del desarrollo de un sitio Web, debemos de tener en mente que una parte medular es el Frontend, y que su diseño debe de ser separado por completo del funcionamiento del sistema. Un Frontend se conforma por código HTML, Hojas de Estilo y JavaScript, de manera general esos serian los principales elementos que contiene cualquier pagina Web. La idea de separar el Frontend del resto del sistema parte de que cada navegador Web cuenta con su propio motor de

navegación, lo que implica que no hay una uniformidad entre un navegador y otro, por tanto podría provocar que una etiqueta HTML no sea correctamente interpretada, marcar un error de JavaScript, o ignorar la Hoja de Estilos, lo cual refleja un grave problema al desarrollar un sitio Web, si no se siguen los estándares que marca el W3C.<sup>1</sup>

El desarrollo de un Frontend presenta varios retos a vencer, desde la correcta interpretación del código HTML y JavaScript en los diferentes navegadores Web que existen en el mercado, hasta las diferentes características de los dispositivos donde será desplegado.

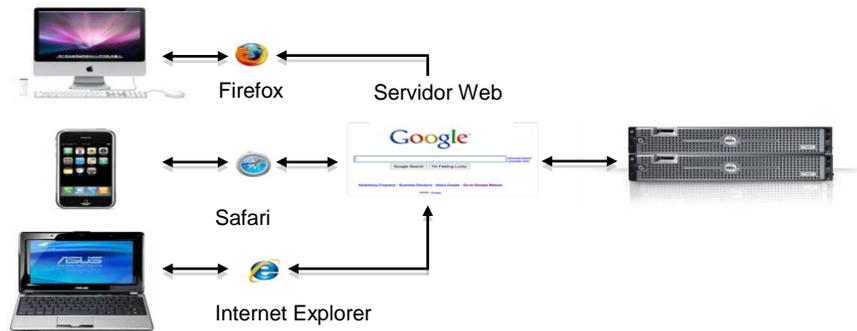


fig. II.5 Plataforma Cliente – Servidor

En la fig. II.5 podemos observar como a través de distintos dispositivos con distintas capacidades de procesamiento y con diferentes tamaños de pantalla, por medio de un navegador Web, ya sea Firefox, Safari ó Internet Explorer, se hace una solicitud a través de la URL ([www.google.com](http://www.google.com)), la cual es atendida por el servidor Web, por lo que responde devolviendo código HTML, código JavaScript y hojas de estilo (CSS) que son recibidas por el navegador Web para ser interpretadas y después desplegar en pantalla la página Web como resultado a la solicitud hecha al servidor Web. En este esquema nuestros clientes son los distintos dispositivos en conjunto con los diferentes navegadores Web por lo que nuestro servidor Web es el que recibe y procesa las solicitudes de los clientes. A este esquema se le conoce como Cliente-Servidor.

## Problemas de compatibilidad

La compañías fabricantes de navegadores Web como Mozilla, Apple, Microsoft entre otras provocan graves problemas para los programadores de páginas Web, ya que, aunque todos los navegadores utilizan JavaScript como uno de los lenguajes de programación, los objetos no se comportan de la misma forma, lo que obliga con frecuencia a programar las páginas en más de una versión, una para Firefox, otra para Safari y una más para Internet Explorer, etc.; en suma, no todos lo navegadores se comportan idénticamente, incluso las diferentes versiones de un mismo navegador se comportan de forma distinta.

<sup>1</sup> W3C - *World Wide Web Consortium*. es una comunidad internacional que desarrolla estándares para garantizar el crecimiento a largo plazo de la Web. <http://www.w3.org/>

El W3C, el consorcio encargado de definir los estándares de la Web, guiado de la mano de Tim Berners-Lee<sup>2</sup> creador del HTML, decidió crear un modelo de objetos único, el DOM, para que todos los fabricantes de navegadores pudieran adoptarlo, facilitando la compatibilidad plena entre ellos.

No obstante, Microsoft ha añadido su propia extensión al DOM, Firefox usa su propio motor llamado Gecko<sup>3</sup>, y Safari usa Webkit<sup>4</sup>. Lo que crea problemas de interoperabilidad entre los navegadores Web

Internet Explorer, es el navegador Web más utilizado por los internautas, esto lleva a un problema real a los fabricantes de navegadores más comprometidos con los estándares, como Mozilla con su Firefox. Si adoptan las extensiones de Microsoft al DOM, se arriesgan a perder credibilidad en sus llamadas a los sitios Web que respetan el estándar, y si no lo hacen, se arriesgan a alejar a sus usuarios por la pérdida de compatibilidad con casi todos los sitios Web que utilizan las extensiones de Microsoft.

Esto sólo cambiaría si los navegadores que respetan los estándares ganan una cuota de mercado significativa en la Web, de forma que el uso de extensiones no estándares se convierta en un problema comercial para los autores de los sitios Web que las usen. Como ya esta pasando con Mozilla Firefox, el cual desde hace varios años, viene ganando terreno en el mercado en comparativa con su rival Internet Explorer de Microsoft.

## Usabilidad

Una vez superados los retos de compatibilidad entre navegadores Web, hay que tener presente que la Usabilidad es otro punto imprescindible ya que se destaca como atributo fundamental para el éxito de un producto de Software.

La Organización Internacional para la Estandarización (ISO) ofrece dos definiciones de Usabilidad:

### **ISO/IEC 9126:**

*"un conjunto de atributos que influyen en el esfuerzo necesario para su uso, y en la evaluación individual de tal uso"*<sup>5</sup>

### **ISO/IEC 9241:**

*"La medida en que un producto puede ser usado por usuarios específicos para alcanzar objetivos específicos con eficacia, eficiencia y satisfacción en un contexto de uso específico."*<sup>6</sup>

*"En general, la usabilidad se refiere a cuán bien los usuarios pueden aprender a utilizar un producto para alcanzar sus objetivos y que tan satisfechos están con ese proceso."*<sup>7</sup>

---

<sup>2</sup> Tim Bernes-Lee Graduado de la universidad de Oxford, Inglaterra en 1989. Inventor de la World Wide Web.

<http://www.w3.org/People/Berners-Lee/>

<sup>3</sup> Motor del navegador Web Firefox (Mozilla) <https://developer.mozilla.org/en/Gecko>

<sup>4</sup> Motor del navegador Web Safari(Apple) <http://webkit.org/>

<sup>5</sup> ISO/IEC 9126

[http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=39752](http://www.iso.org/iso/catalogue_detail.htm?csnumber=39752)

[http://www.usabilitynet.org/tools/r\\_international.htm#9126-1](http://www.usabilitynet.org/tools/r_international.htm#9126-1)

<sup>6</sup> ISO/IEC 9241

[http://www.usabilitynet.org/tools/r\\_international.htm#9241-11](http://www.usabilitynet.org/tools/r_international.htm#9241-11)

<sup>7</sup> <http://www.usability.gov/basics/index.html>

## Beneficios de la Usabilidad

- Reducción en los costos de aprendizaje y entrenamiento.
- Disminución de los costos de soporte al usuario.
- Optimización de los costos de diseño, rediseño y mantenimiento.
- Aumento de la tasa de conversión de visitantes a clientes de un sitio Web.
- Mejora la imagen y el prestigio.
- Mejora la calidad de vida de los usuarios, ya que reduce su estrés, incrementa la satisfacción y la productividad.

Un buen sitio de referencia para comenzar a desarrollar sitios Web usables y útiles es: <http://www.usability.gov/> sitio que proporciona una serie de lineamientos para el diseño orientado al usuario y que es manejado por el gobierno de los Estados Unidos de America.

## Componentes de un Front End

Como ya lo habíamos mencionado antes, un Frontend se compone esencialmente de tres elementos que son; HTML, JavaScript y CSS, claro que estos no son de manera definitiva todos los elementos ya que hay otro tipo de elementos que usan otras tecnologías como lo puede ser una applet de Java ó un archivo Flash embebido en el código HTML, entre otros; para fines prácticos solo abarcaremos HTML, JavaScript, AJAX y CSS.

## HTML

Este trabajo no tiene la finalidad de dar a conocer todas las bondades de HTML, así que de manera practica se abarcará lo esencial para entender su funcionamiento. En caso de que el lector este interesado en extender este tema puede consultar el siguiente sitio Web de referencia: <http://www.w3.org/html/>

HTML por sus siglas en ingles HyperText Markup Language (*Lenguaje de Marcas de Hipertexto*) consta de una serie de etiquetas que nos sirven para agregar elementos (textos y gráficos) a una página Web, su uso es muy sencillo, veamos un ejemplo:

```
<title> Titulo de la Página Web </title>
```

En este ejemplo hacemos uso de la Etiqueta(tag en ingles) “title” para agregar el titulo de nuestra página Web, que comúnmente aparece en la parte superior de nuestro navegador, en este ejemplo vemos que iniciamos con la etiqueta de apertura “<title>” y terminamos de hacer uso con la etiqueta de clausura “</title>”, esto quiere decir que el texto que se agregue entre la etiqueta de apertura y entre la etiqueta de clausura será tomado por el navegador Web, y colocado en la parte superior del mismo.

Teniendo claro el ejemplo anterior veamos otro ejemplo un poco más completo:

```

1 <html lang="es">
2   <head>
3     <title> Titulo de la Página Web </title>
4   </head>
5   <body>
6     <strong>Hola Mundo</strong>
7   </body>
8 </html>

```

Fig. II.6 Contenido del archivo ejemplo.html



Fig. II.7 Resultado del archivo ejemplo.html

En la Fig. II.6 vemos el código fuente de la página Web mostrada en la Fig. II.7, nos podemos percatar que HTML es una colección de etiquetas las cuales nos permite agregar elementos además de darnos las herramientas para manipular el contenido en nuestra página Web. Al guardar el contenido del archivo (Fig. II.6) en un archivo llamado “ejemplo” con la extensión .html o .htm y después abrirlo con algún navegador Web podremos ver los resultados (Fig. II.7)

HTML es un lenguaje de etiquetas que nos permite agregar elementos a nuestra página Web, tanto textos como imágenes, para que sean guardados en un archivo para luego ser interpretados en algún navegador Web. Hay algunas cosas que hay que recalcar, como ya lo habíamos mencionado cada navegador Web tiene su propio motor para interpretar el HTML, por lo que para evitar problemas de compatibilidad será necesario apegarse a los estándares de la W3C, por otra parte como ya lo vimos en nuestro ejemplo (Fig. II.6 y Fig. II.7) HTML solo nos permite crear elementos en nuestra página Web, sin embargo no tienen la capacidad de procesar información es por eso que no se le considera un lenguaje de programación.

¿Entonces HTML podría resolver una simple operación aritmética (2+2)? La respuesta a esa pregunta es No, por si mismo HTML no tiene las capacidades de procesar datos, para realizar esto se necesitaría la ayuda de algún lenguaje de programación como PHP, Java, C#, vistos en el capítulo uno.

El W3C(World Wide Web Consortium) es el encargado de formular la especificación de HTML que servirá de estándar para los navegadores Web. Por lo que también nos proporciona un sistema de validación de HTML muy útil para evitar caer en problemas de compatibilidad. Esta herramienta esta disponible desde la siguiente dirección de Internet:

<http://validator.w3.org/>

## JavaScript

JavaScript es un lenguaje de scripts, interpretado, multiplataforma y parcialmente orientado a objetos. Fue creado por Netscape específicamente para su uso en el desarrollo de sitios Web. Actualmente es un estándar mantenido por el ECMA<sup>8</sup>.

<sup>8</sup> ECMA Asociación dedicada a la estandarización de información y sistemas de comunicación. <http://www.ecma-international.org/>

El código JavaScript puede enlazarse o añadirse a las páginas Web proporcionando un control total y dinámico sobre ellas. Además, también permite controlar -hasta cierto punto- las aplicaciones que lo ejecutan, habitualmente navegadores. Veamos el siguiente ejemplo:

```

1 <html lang="es">
2 <head>
3 <title></title>
4 </head>
5 <body>
6 <script language="javascript">
7     document.title="Este es el titulo de la pagina";
8     document.write("HOLA MUNDO");
9 </script>
10 </body>
11 </html>

```

Fig. II.8 Contenido del archivo ejemplo2.html

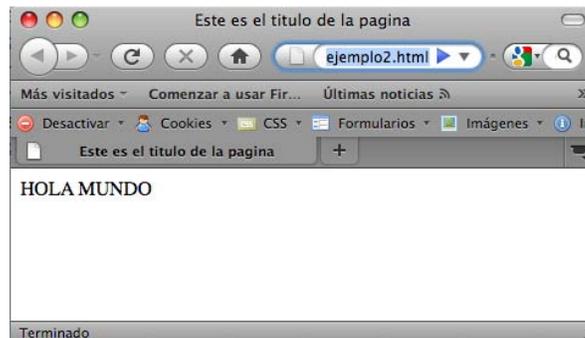


Fig. II.9 Resultado del archivo ejemplo2.html

Como podemos apreciar en la Fig. II.8 seguimos teniendo el mismo código HTML mostrado en la Fig. II.6 pero ahora hemos agregado un script en lenguaje JavaScript (de la línea 6 a la línea 9) para demostrar la forma en que se pueden manipular los elementos HTML por medio de este lenguaje. En la línea 7 del código tenemos una instrucción en JavaScript para asignarle un valor al elemento “title” de HTML. En la línea 8 tenemos otra instrucción pero ahora para imprimir en pantalla el texto “HOLA MUNDO”

JavaScript además de permitirnos modificar los valores y propiedades de los elementos HTML, también nos da la capacidad de procesamiento, por ejemplo podríamos realizar una calculadora en este lenguaje.

## AJAX

Por sus siglas en inglés significa Asynchronous JavaScript And XML, es una tecnología muy popular hoy en día y es una característica fundamental en el desarrollo de Webs 2.0 por que permite la comunicación asíncrona entre el cliente y el servidor, esto nos abre las puertas a muchos beneficios, por ejemplo, cuando uno hace una búsqueda Web en tiempo real con Google (<http://www.google.com/webhp?esrch=RTSearch>) nos muestra una sección llamada “Lastest Results” la cual se muestra como si fuera un display en el que van apareciendo las ultimas coincidencias encontradas de nuestras palabras clave. En otras palabras es una actualización en tiempo real de lo último encontrado, sin la necesidad de que el usuario tenga que hacer una nueva búsqueda. AJAX es la tecnología que permite que cada determinado tiempo se actualice esta sección de la página de manera asíncrona sin necesidad de que el usuario tenga que dar clic en el botón de recargar en su navegador para volver a mandar a traer toda la información desde el servidor, sino que esta tecnología se encarga de traer explícitamente la información que se va agregando al sitio permitiendo tener una mejor experiencia de navegación y con un buen manejo de la tecnología nos permite disminuir el tráfico de la red además de liberar de carga de trabajo al servidor.

*“Actualmente, la creación de aplicaciones Web resulta un proceso pesado y propenso a errores. Los desarrolladores pueden pasar el 90% de su tiempo estudiando las peculiaridades de los navegadores. Por otra parte, la creación, la reutilización y el mantenimiento de una gran cantidad de componentes AJAX y bases de código JavaScript pueden ser tareas complejas y delicadas.”<sup>9</sup>*

Existen actualmente varios Frameworks para el manejo de JavaScript los cuales nos facilitan mucho el uso de esta tecnología además de proporcionarnos muchas herramientas listas para usarse. A continuación se enlistan algunos de los Frameworks mas usados por desarrolladores Web.

- Prototype
  - <http://www.prototypejs.org/>
- jQuery
  - <http://jquery.com/>
- Mootools
  - <http://mootools.net/>
- Script.aculo.us
  - <http://script.aculo.us/>
- Google Web Toolkit
  - <http://code.google.com/intl/es/webtoolkit/>
- Yahoo Yui Library
  - <http://developer.yahoo.com/yui/>
- Adobe Spry
  - <http://labs.adobe.com/technologies/spry/>

## Cloud Computing

La computación en la nube es un concepto general que incorpora el software como servicio<sup>10</sup>, tal como la Web 2.0, como ejemplos de Computación en la Nube destacan Amazon EC2, Google Apps<sup>11</sup> y Microsoft Azure<sup>12</sup>, que proveen aplicaciones comunes de negocios en línea accesibles desde un navegador Web, mientras el software y los datos se almacenan en los servidores.

Actualmente compañías como Google y su nuevo sistema operativo Google Chrome OS, tienen la finalidad de explotar las características del Cloud Computing de tal forma que a mediados del año 2010 se comercializarán computadoras sin disco duro las cuales estarán obligadas a contar con una conexión a Internet para poder acceder a las aplicaciones y a los datos que serán almacenados en la nube.

AJAX bien puede ser una tecnología que apoye en estas labores, ya que como anteriormente hemos mencionado, basado en un esquema de cliente - servidor nos permite tener una comunicación asíncrona. Veamos el siguiente ejemplo de lo que se puede lograr hacer con AJAX.

---

<sup>9</sup> Google Web Toolkit. <http://code.google.com/intl/es/webtoolkit/>

<sup>10</sup> Amazon EC2 <http://aws.amazon.com/ec2/>

<sup>11</sup> Google Apps <http://www.google.com/apps/>

<sup>12</sup> Microsoft Azure <http://www.microsoft.com/windowsazure/>

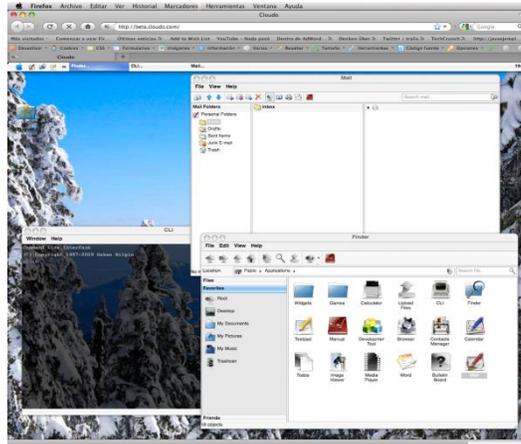


Fig. II.10 Sistema Operativo Web – Cloudo



Fig. II.11 Sistema Operativo Web - Cloudo

En la Fig. II.10 y Fig. II.11 podemos apreciar una captura de pantalla, donde se puede distinguir que por medio de nuestro navegador Web (Firefox) ingresamos a la dirección de Internet [www.cloudo.com](http://www.cloudo.com) donde previamente nos identificamos para tener acceso al sistema que esta construido bajo una interfaz Web usando AJAX imitando un sistema operativo de escritorio que te permite tener acceso a tu lector de correo, un explorador de archivos, una agenda, además de poder configurar el tema ó el fondo de escritorio. Definitivamente esto es un vistazo de lo que Google Inc. nos pretende ofrecer en la nueva generación de computadoras sin disco duro y con Google Chrome Os instalado.<sup>13</sup>

<sup>13</sup> El Universal - Google Chrome OS  
<http://www.eluniversal.com.mx/notas/641120.html>  
<http://www.eluniversal.com.mx/articulos/56641.html>

## CSS

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los *Estilos* definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

```

1 <html>
2 <head>
3 <title>Ejemplo de uso de CSS</title>
4 <style type="text/css">
5
6     body {
7         padding-left: 11em;
8         font-family: Georgia, "Times New Roman", serif;
9         color: red;
10        background-color: #d8da3d;
11    }
12
13    h1 {
14        font-family: Helvetica, Geneva, Arial, sans-serif;
15    }
16
17 </style>
18 </head>
19 <body>
20 <h1>HOLA MUNDO</h1>
21 </body>
22 </html>

```

Fig. II.12 Contenido del archivo ejemplo3.html



Fig. II.13 Resultado visual del archivo ejemplo3.html

Tomando como ejemplo el archivo “ejemplo3.html” (Fig. II.12) podemos denotar el uso de CSS de la línea 4 hasta la 17. Las especificaciones generales de nuestra página HTML como margen izquierdo, tipo de fuente, color de fuente y color de fondo se encuentran de la línea 6 a la 11. Y por último la personalización de las etiquetas HTML <h1> de la línea 13 a la 15 donde se indica el tipo de letra para la etiqueta.

## Ventajas al utilizar CSS

- Diseño centralizado lo que se agiliza de forma considerable los cambios en el sitio Web.
- Una página puede disponer de diferentes hojas de estilo según su uso, por ejemplo, para ser impresa o mostrada en un dispositivo móvil.
- Un documento HTML es más claro de entender y se consigue reducir considerablemente su tamaño. Si los CSS son colocados en un archivo por separado. Lo que a su vez se refleja en mayor visibilidad para los robots<sup>14</sup>, permitiendo tener un mejor posicionamiento en los motores de búsqueda.

Un buen sitio de referencia para poder visualizar el potencial de CSS es:

<http://www.csszengarden.com/>

El W3C(World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los navegadores Web. Por lo que también nos proporciona un sistema de validación de CSS muy útil para evitar caer en problemas de compatibilidad. Esta herramienta esta disponible desde la siguiente dirección de Internet:

<http://jigsaw.w3.org/css-validator/>

---

<sup>14</sup> Programa computacional encargado de buscar información en la red de manera metódica y automatizada.

# Backend



CAPITULO III

El complemento al Frontend se le conoce como Backend, mientras que el Frontend se refiere a la interfaz gráfica de usuario, el Backend se encarga de las tareas de mantenimiento, por ejemplo imaginemos que nos encontramos en un sitio de noticias, nosotros como internautas podremos navegar por la variedad del contenido, entre videos, imágenes, notas y reportajes para mantenernos informados, pero mientras navegamos por el sitio nos vamos a percatar que de un instante a otro ya existe una nueva nota de último momento con la información más reciente, algo que es muy valioso para estar informado.

¿Cómo llego esa información ahí?

Bueno es simple, mientras los usuarios interactúan con el Frontend, habrá editores interactuando con el Backend agregando nuevas notas y reportajes, de esa forma existirá información actualizada, nuevas imágenes y videos que los usuarios podrán consultar en línea.

Un Backend hace que un sitio sea funcional y robusto, cuidando los detalles detrás de escena, realizando tareas que no son directamente controladas por el usuario final.

Por lo tanto un Backend es aquel que se encarga de las tareas de mantenimiento, actualización, configuración de un sitio Web. Además de interactuar directamente con el sistema gestor de base de datos(SGBD) para hacer consultas o para insertar nueva información.

## **CRUD**

Continuando con nuestro ejemplo del sitio de noticias, podemos llegar a resumir que existen cuatro operaciones básicas que se le pueden aplicar a cierta información:

Create: inserción de nueva información que será almacenada en la base de datos.

Read: operación solo de lectura de la información, aquí solo se consulta el dato en base de datos, sin posibilidad de afectarlo.

Update: actualización de la información, es una operación compuesta, donde primero se consulta el dato para después editarlo y almacenarlo.

Delete: operación de borrado de una parte o de la totalidad de la información.

## **Niveles de Usuario**

Es común que un Backend tenga distintos niveles de usuarios, para tener acceso a diferente tipo de información. Por ejemplo podemos tener a un Administrador que tiene todos los permisos, pero de igual manera tiene toda la responsabilidad, mientras que un editor podría solo tener permisos para editar cierta información, por que cualquier error que cometiera, no perjudicaría de manera crítica al sitio, por consecuencia la responsabilidad que carga un editor es mucho menor a la de un administrador.

De esta manera se pueden otorgar diferentes niveles de acceso al personal autorizado, aumentando la seguridad y la estabilidad evitando que posibles errores comunes se cometan y provoquen que la información no se despliegue correctamente, incluso se puede prevenir una caída del servidor Web.

## Wordpress<sup>1</sup>

Es un CMS por sus siglas en ingles, Sistema Manejador de Contenido, tiene la finalidad de facilitarnos las herramientas para el fácil manejo y mantenimiento de un sitio.

Wordpress esta conformado por dos elementos clave:

**Frontend:** donde cualquier usuario puede consultar información con libertad por medio de la URL del sitio. Aunque en algunos sitios se suele requerir que el usuario se registre y se identifique para tener acceso a cierta información, no es un requerimiento primordial, más bien tiene la finalidad de recavar datos importantes de los usuarios como lo son el correo electrónico y nombre . Por otra parte esos datos también sirven para generar estadísticas y saber cual es la información de mayor relevancia, lo cual puede ayudar a enviar una campaña publicitaria dirigida de acuerdo a los intereses de cada usuario.

**Backend:** panel administrador del contenido del sitio, al cual solo tiene acceso cierto personal autorizado. El usuario se tiene que identificar (login) para poder acceder al sistema y de acuerdo a su nivel de acceso tendrá una cantidad limitada de posibles acciones, como lo es editar una publicación, agregar una nueva imagen, borrar un comentario, o simplemente leer cierta información antes de su publicación. Las cuatro posibles operaciones básicas que se le pueden aplicar a un dato almacenado en una base de datos.

Una imagen dice más que mil palabras, es por eso que ahora lo vamos a distinguir de manera gráfica para ratificar lo anteriormente mencionado.



fig III.1 Frontend sitio ewiko.com.mx

<sup>1</sup> Wordpress Sitio Oficial <http://wordpress.org/>

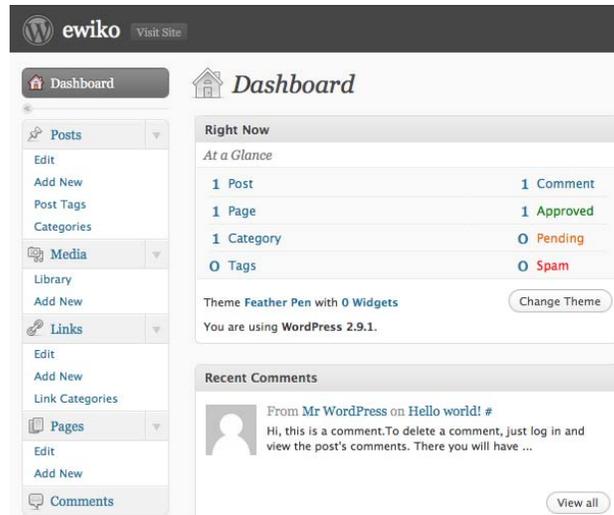


fig. III.2 Backend sitio ewiko.com.mx

En el Backend (fig. III.2), encontramos todas las opciones disponibles dependiendo de tu nivel de acceso (editor, autor, administrador) y es aquí donde el usuario puede fabricar el contenido que será posteriormente exhibido en el Frontend.

Por ejemplo, el autor podría redactar un nuevo artículo, guardarlo sin publicarlo, para regresar más tarde a leerlo y modificarlo antes de su publicación. Y en caso de que sea necesario un administrador podría entrar a borrarlo.

Tenemos que distinguir que el Frontend no sufrirá ningún tipo de cambio visible, hasta que no sea aplicado el cambio desde el Backend.

Tanto el Frontend como el Backend son complemento uno del otro, y trabajan de manera conjunta para hacer mucho más práctico para el administrador del sitio las tareas rutinarias (CRUD).

El Backend constituye la columna vertebral de un sitio y es indispensable para que tenga una vida longeva, evitando que la edición de páginas estáticas se vuelva una tarea imposible al paso del tiempo mientras el contenido va creciendo, acelerando la actualización de la información.

## Base de Datos

Una base de datos es un recurso indispensable en el desarrollo de software, ya que permite el almacenamiento de datos organizados y relacionados entre sí, para más tarde ser consultados, modificados o eliminados por medio de algún sistema de información.

### Ventajas:

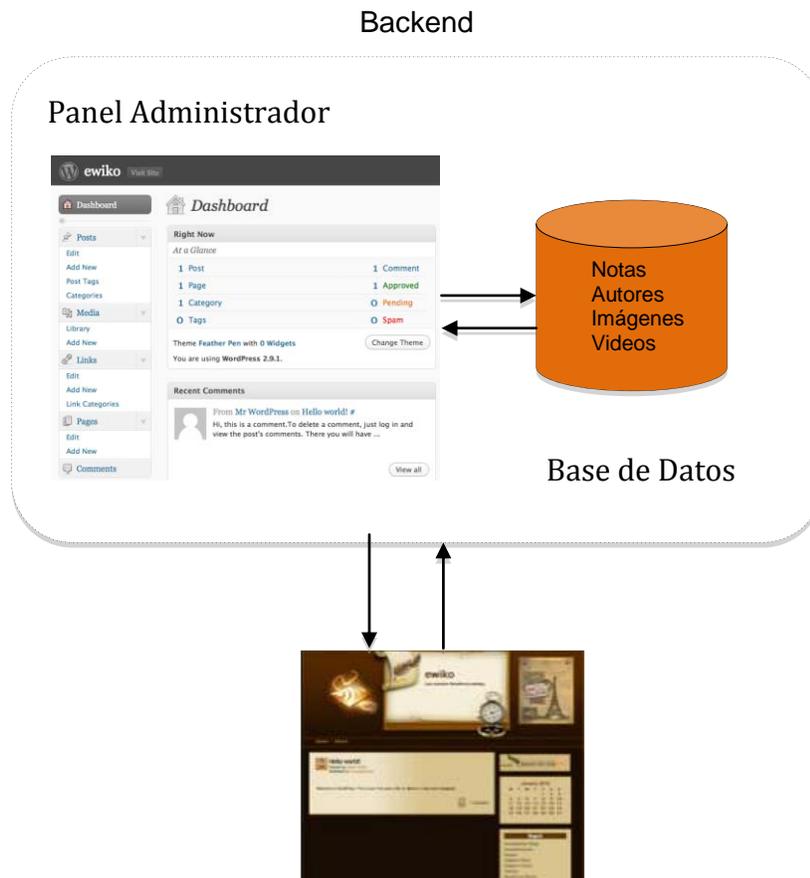
- Redundancia mínima
  - La finalidad es que los datos no se repitan ó al menos que se repitan lo menos posible.
- Acceso concurrente
  - Permite la conexión simultánea de varios usuarios a la misma base de datos.

- Conexión a través de lenguajes de programación
  - Lo que genera una gran ventaja, por que al poder manipular la base de datos por medio de algún lenguaje de programación nos permite automatizar muchas tareas.
- Facilidad de Consulta
  - Al manejar grandes cantidades de información una base de datos facilita la consulta de datos reduciendo el tiempo de búsqueda.

## MySQL<sup>2</sup>

Es el sistema gestor de base de datos más popular dentro del ámbito del open source, con más de 100 millones de copias distribuidas a través de su trayectoria. MySQL ha llegado a ser preferida para el desarrollo de sitios Web, grandes compañías como Google, Yahoo y Nokia hacen uso de esta exitosa base de datos.

Una base de datos forma parte de un Backend, interactuando constantemente realizando las operaciones de CRUD. La fig. III.3 ilustra esta idea.



<sup>2</sup> MySQL Sitio Oficial <http://www.mysql.com/about/>

La fig. III.3 nos muestra como es que la base de datos puede tener comunicación con el Backend, para almacenar la información del sistema. En nuestro ejemplo vemos que podríamos almacenar los datos del autor que escribe en nuestro portal de noticias, por supuesto que también podemos almacenar la información de la misma nota, incluyendo imágenes y videos.

“Autor” podría ser el nombre de una tabla en la base de datos, la cual nos serviría para guardar los datos de cada uno de los colaboradores de nuestro portal, y de esta forma almacenar la información centralizada y organizada, obteniendo como beneficio consultas rápidas y eficientes.

Vamos a desglosar la tabla “Autor”, ¿qué datos son los que debería de contener?, Esto lo podemos ilustrar de la siguiente manera.

TABLA AUTOR				
idAutor	nombre	apMaterno	apPaterno	email
1	Pablo	Rodriguez	Guerrero	<a href="mailto:pablo@gmail.com">pablo@gmail.com</a>
2	Erika	Flores	Cristiani	<a href="mailto:erika@gmail.com">erika@gmail.com</a>
3	otro	registro	mas	<a href="mailto:email@gmail.com">email@gmail.com</a>
4	...	...	...	...
5	...	...	...	...

fig. III.4

Esta tabla (fig. III.4) se puede visualizar como un cajón dentro de un gran archivero, dentro de este cajón solo colocaremos la información a fin, pero en el resto de los cajones del archivero podemos organizar el resto de la información, una base de datos es un archivero con “N” cantidad de cajones.

### ¿Cómo se relacionan estos datos?

Un Autor es un colaborador que contribuye con notas y puede ser que escriba una o varias notas. Entonces concluimos que un Autor esta relacionado con distintas notas.

TABLA NOTA				
idNota	idAutor	titulo	texto	fecha
1	2	Liguilla	abc ...	15-ene-10
2	3	NFL	abc...	10-ene-10
3	1	Elecciones	abc...	05-jul-09
4	1	Clima	abc...	22-dic-09

Fig III.5

En la tabla “Nota” (fig III.5) nos podemos percatar que tenemos una columna llamada idAutor, esto es por que todos los datos del autor ya los tenemos almacenados en la tabla “Autor” (fig III.4), por lo que sería redundante volver a tener que agregar esta información. En su lugar tenemos un numero que se refiere al identificador(ID) del autor.

Por ejemplo, la nota numero 3(IdNota), tiene asignado el autor numero 1(idAutor), por lo cual si revisamos la fig. III.4 nos daremos cuenta que se refiere al autor “Pablo Rodríguez Guerrero”. Este mismo autor esta relacionado con la nota numero 4, sin embargo no repetimos la información del Autor, más bien la tenemos centralizada dentro de un solo registro en una única tabla . A este tipo de Base de Datos se les conoce como relacionales.

## SQL

Es el lenguaje estándar marcado por el American National Standards Institute<sup>3</sup> para acceder y manipular a base de datos, como por ejemplo, MySQL, SQL Server, Access, Oracle, Sybase, DB2 y otros sistemas gestores de base de datos (SGBD).

Algunas de las acciones que te permite hacer el lenguaje SQL sobre una Base de Datos, son las siguientes:

- Consultar la información en la base de datos
- Insertar nuevos registros a la base de datos
- Actualizar registros en la base de datos
- Eliminar registros de la base de datos
- Crear una nueva base de datos
- Crear nuevas tablas en la base de datos

Continuando con nuestro ejemplo y nuestras tablas “Autor” y “Nota”, veremos un ejemplo en el lenguaje SQL para obtener por medio de una consulta a la tabla “Nota”, todos los registros que le correspondan al autor “Pablo Rodríguez Guerrero”.

```
1 SELECT * FROM NOTA WHERE idAutor = '1';
2
3
```

fig. III.6

La fig. III.6 muestra de manera intuitiva y de forma muy sencilla, como por medio de una consulta en lenguaje SQL y en una sola línea de código podemos lograr nuestro objetivo. Previamente nosotros ya sabemos que autor “Pablo Rodríguez Guerrero” tiene el idAutor numero 1, por lo que nos facilita la consulta. Como resultado obtenemos lo siguiente.

TABLA NOTA					
idNota	idAutor	titulo	texto	fecha	
3	1	Elecciones	abc...	05-jul-09	
4	1	Clima	abc...	22-dic-09	

fig. III.7

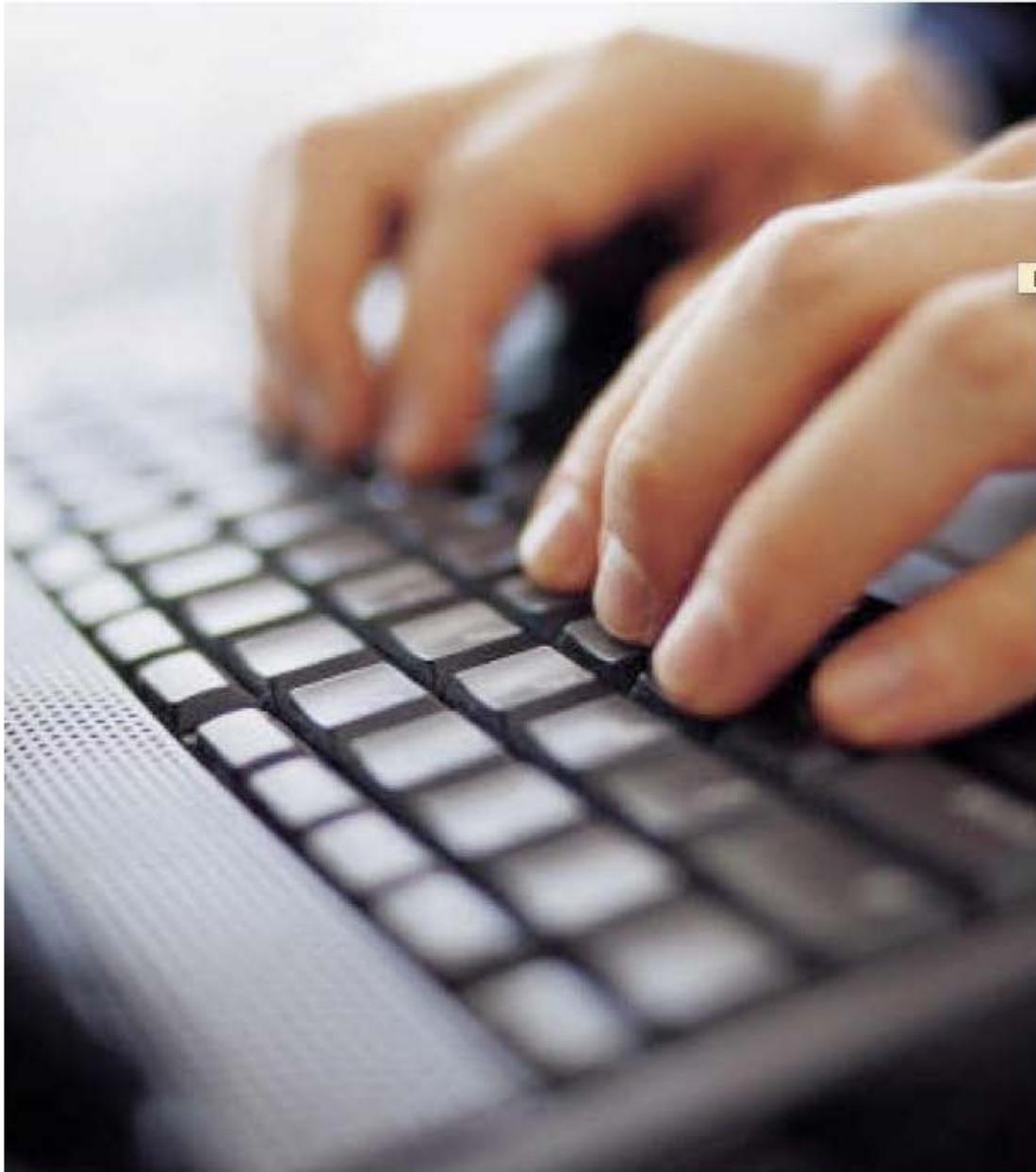
El tema de Base de Datos por si mismo es un tema bastante extenso, por lo que sale de los objetivos de este trabajo. De manera objetiva se ha explicado el funcionamiento de una Base de Datos y el uso del Lenguaje SQL.

<sup>3</sup> ANSI es una organización sin ánimo de lucro que supervisa el desarrollo de estándares.

Para concluir este capítulo y lo referente a un Backend, podemos resaltar su importancia para que un sitio se convierta en un sistema escalable y de esta manera garantizar varios años en producción. Como miembro inherente tenemos al Sistema Gestor de Base de Datos, ya que es clave para la administración inteligente de la información, basándonos en el lenguaje de facto SQL para manipular Base de Datos podremos automatizar tareas rutinarias como el CRUD.

Por simple lógica se intuye que el Backend se encuentra accesible para aquellas personas relacionadas con la colaboración y administración del Sitio Web, permaneciendo de forma inaccesible para el usuario final.

# Wizard



Est

CAPITULO IV

## Metaprogramación

De manera sencilla podemos describir a la Metaprogramación como “programas que fabrican programas”.

“La Metaprogramación consiste en escribir código que no ataca directamente al dominio del problema que queremos atacar, sino al código que lo resolvería. Dicho sea de otro modo, el código que escribimos no modifica los datos o el estado de nuestra información, sino el del programa.”<sup>1</sup>

Para entenderlo mejor vamos a hacer una analogía, en la película de Matrix, el Sr. Smith es un claro ejemplo de Metaprogramación, ya que el tiene la característica de poderse clonar así mismo e incluso modificar y/o controlar a otros.

Entrando en detalles técnicos podemos citar el siguiente ejemplo:

```

1  <?php
2
3  $str = "class HelloWorld{ \n";
4  $str .= "\t public static void main(String... args){ \n";
5  $str .= "\t\t System.out.println(\"Hello World\"); \n";
6  $str .= "\t } \n";
7  $str .= "}\n";
8
9  $fHandler = fopen("HelloWorld.java", "w+");
10 fputs($fHandler,$str);
11 fclose($fHandler);
12
13 ?>
14

```

Fig. IV.1

En la Fig. IV.1 encontramos un programa escrito en el lenguaje PHP (visto en el capítulo I), pero que a diferencia este código en vez de mandar a imprimir directamente en pantalla el texto “Hola Mundo”, escribe un programa en lenguaje Java y lo guarda en un archivo llamado “HelloWorld.java”, el cual al ser ejecutado manda a imprimir a pantalla la frase “Hello World”, entonces mediante un programa en lenguaje PHP fabricamos otro en lenguaje Java.

Aunque el ejemplo es bastante sencillo, hay que destacar el concepto de la Metaprogramación, donde un programa fabrica otro programa, y para esto el programador no debe de pensar en resolver un problema aislado, si no, mas bien debe de pensar en solucionar un conjunto de problemas de un dominio específico. Un concepto semejante al de un Framework (visto en el capítulo I).

Después de ejecutar el código de la Fig. IV.1 se creará un archivo llamado “HelloWorld.java” a continuación se muestra su contenido.

```

class HelloWorld{
    public static void main(String... args){
        System.out.println("Hello World");
    }
}

```

Fig. IV.2

<sup>1</sup> Revista en línea SG Software Guru <http://www.sg.com.mx/content/view/718/1/>

La Fig. IV.2 es un programa escrito en lenguaje Java, resultado de ejecutar el programa escrito en lenguaje PHP (Fig. IV.1).

Si trasladamos este mismo concepto a un nivel un poco más complejo, como por ejemplo un Backend que fabrica otro Backend, ya se escucha mucho más interesante.

Dentro de la definición de los lenguajes de cuarta generación, se tienen contemplados a los generadores de código por lo que un Wizard entra dentro de esta definición ya que por medio de él podemos de manera asistida generar aplicaciones.

Precisamente un Wizard que nos asista de manera gráfica y que nos permita generar un Backend sin necesidad de escribir ni una sola línea de código es el tema central de este proyecto.

## **Scaffolding**

Esta es una técnica usada por los Frameworks actuales como Grails para Java ó CakePHP y Symfony para PHP, entre otros.

Consiste en generar código para automatizar las tareas de CRUD para cualquier tabla en la base de datos. Por ejemplo si tenemos una tabla “usuarios” con los campos nombre, dirección y teléfono, por medio de esta herramienta podremos acceder casi de manera inmediata a una interfaz que nos permita ingresar nuevos registros, actualizar, visualizar y borrar. Lo cual nos serviría en tiempo de desarrollo para hacer pruebas y en tiempo de producción para realizar tareas rutinarias.

El Scaffolding es posible a partir de un Framework, ya que nos provee las herramientas que serán reutilizadas una y otra vez para un comportamiento claramente definido como lo es el CRUD para un registro de base de datos. Desafortunadamente este no es la herramienta que nos ayudara a resolver la problemática, ya que no abarca la funcionalidad de Catalogo ni Búsqueda las cuales son fundamentales para un Backend.

## **Generación de Código**

Hemos dicho que el objetivo de este proyecto es la generación automatizada de código que nos permita acceder de manera casi inmediata a un Panel Administrador(Backend) partiendo de la estructura de la base de datos, para lo cual debemos de tener en cuenta los siguientes aspectos fundamentales que lo definen:

- ✓ Manejo de Sesiones:
  - Mejor conocido como Login, donde uno tiene que ingresar su usuario y contraseña para acceder a al sistema.
- ✓ Usabilidad:
  - Interfaz grafica de usuario unificada y de fácil uso para el usuario final.
- ✓ Niveles de usuario:
  - Un menú de opciones acorde al nivel de usuario es indispensable en cualquier Backend.
- ✓ Validación de Campos:
  - Cuando ingresamos registros a la base de datos es necesario validar la información para verificar que realmente es lo que se requiere para conservar una consistencia en los datos.

- ✓ **Catálogo:**
  - Este será fundamental para mostrar de manera ordenada los registros de la base de datos, además de darnos la opción de búsqueda y enlazar por medio de un identificador(ID) para poder visualizar los detalles del registro, editarlo y quizás borrarlo.
- ✓ **Paginación de Resultados:**
  - Cuando mostramos resultados de búsqueda o un listado de información es importante recurrir a la paginación que nos permite entregar solo una cierta cantidad de registros por página y no saturar al usuario final con toda la información en una sola página.
- ✓ **Mensajes de Sistema:**
  - En cualquier momento será pertinente contar con una interfaz de mensajes para comunicarle al usuario final mensajes de aceptación, error o de prevención.
- ✓ **Búsqueda:**
  - Hoy en día se manejan una cantidad incontable de información por lo que convierte a este módulo como indispensable para un Backend.
- ✓ **Jerarquía de Directorios:**
  - La cual nos permite tener de forma organizada nuestras Modelos, Vistas y Controladores además nuestras Hojas de estilo (CSS), Imágenes y archivos JavaScript requeridos.
- ✓ **Archivo de configuración:**
  - En el cual podemos agregar configuraciones globales como lo son las rutas de directorios, definición de constantes, e incluso el usuario, contraseña, y nombre de la base de datos a la cual se conectará nuestra aplicación.
- ✓ **Utilerías:**
  - No siempre es necesario construir una Clase. Cuando se tiene un solo método, será mejor tener una colección de ellos agrupados en un archivo para ser usados cuando se requiera. Podemos tener utilerías para pruebas, formateo de cadenas, expresiones regulares, entre otras.

## Dominio del Problema

Se requiere realizar un Wizard basado en un Framework para que de manera asistida nos permita generar código para automatizar las tareas rutinarias del CRUD para cada una de las tablas de la base de datos a la cual se conecta el sistema, lo cual ahorrará una gran cantidad de tiempo en el desarrollo de un Backend y nos permitirá tener como resultado un código limpio y estandarizado que facilite implementar y/o extender la aplicación.

De esa manera el programador podrá utilizar el tiempo ganado para la fase de pruebas y mejoras de interfaz gráfica del usuario final (Frontend). En el capítulo 2 comentamos que una aplicación Web que involucra tecnologías como HTML, CSS, JavaScript tendrá que apegarse a los lineamientos de la W3C adicionalmente de que tendrá que ser probada en los distintos navegadores que existen en el mercado para garantizar su correcto funcionamiento.

## Análisis

Se puede visualizar como una caja negra donde primordialmente se alimenta de la estructura de la base de datos, obteniendo como resultado el código necesario para la manipulación de la información (CRUD).

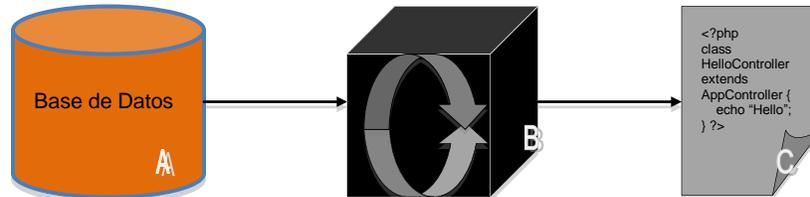


Fig IV.3 Análisis de Solución

Son tres pasos secuenciales que hay que seguir (A,B,C). Por lo que se entiende que todo inicia por la definición de la base de datos, ¿Cuáles serán los entes que compondrán a nuestra aplicación final?, ¿Cuáles son los elementos que conforman a cada ente?, ¿Qué información será indispensable guardarla?. Habrá que hacer un análisis previo para contestar a estas preguntas y definir el esquema de base de datos para así continuar con el siguiente paso.

En el paso B es donde se encuentra el Wizard, sustentado en un Framework que nos ayude al manejo de módulos relevantes al dominio del problema como lo es el manejo de HTML, Sesiones, Validaciones, Catálogos, Búsquedas, entre otros. El paso A servirá para alimentar al Wizard, ya que el Wizard se conectara a la base de datos para conseguir los campos que conforman a cada una de las tablas y de esta manera generar el código de la aplicación final.

En el paso C obtenemos los resultados que nos arroja nuestro Wizard, el cual se encargo a grandes rasgos de parametrizar el comportamiento de CRUD, el cual se conoce a la perfección. Se generan los archivos necesarios y se colocan en un directorio para las primeras pruebas y en caso de estar todo correcto se podrán mover a un nuevo directorio para su permanencia.

## Requerimientos técnicos

Para evitar el pago de licencias se opta por el uso de tecnologías con licencia tipo GNU<sup>2</sup>.

<i>Sistema Operativo</i>	<i>Independiente de plataforma</i>
<i>Lenguaje de Programación</i>	PHP 5.x
<i>Base de Datos</i>	MySQL 5.x
<i>Interfaz Grafica Web</i>	HTML 4.x, CSS, JavaScript
<i>Servidor Web</i>	Apache 2.x

<sup>2</sup> GNU Licencia Publica General <http://www.gnu.org/copyleft/gpl.html>

## Solución

La solución consiste en parametrizar un proceso en el cual ya está bien definido su comportamiento, ya hemos hablado de las constantes. Pero cada vez que se genere un nuevo Backend por medio del Wizard, habrá variables que seguramente van a cambiar de una aplicación a otra.

- ✓ Validación de Campos
  - campos obligatorios
  - campos opcionales
  - no requeridos, sin validación
  
- ✓ Función de Validación
  - email, url, nombre, código postal, tarjeta de crédito, teléfono, numérico, alfanumérico, rango mínimo y máximo de caracteres, RFC, contraseña, número decimal, etc.
- ✓ Etiquetas
  - títulos de campo en catalogo
  - mensajes de validación
  - títulos de campo en formulario
  
- ✓ Despliegue de la Información
  - de acuerdo a los requerimientos de la aplicación podríamos desplegar o no cierto campo. Por ejemplo un campo llamado "ultima\_actualizacion" y que almacene la fecha en formato "timestamp" de la última vez que se actualizó el registro, y que este dato sea agregado de forma automática.
  
- ✓ Ordenamiento
  - dependiendo de las necesidades podríamos listar la información ordenada de forma ascendente o descendente por cierto campo.
  
- ✓ Enfoque de Campo
  - esta es una funcionalidad muy útil a la hora de trabajar repetidamente con la aplicación. Una vez cargada la página se encargará de posicionar el cursor en el campo que le indiquemos, evitándonos usar el ratón.

Veamos un ejemplo para entender mejor la solución, supongamos que tenemos una base de datos llamada ICO<sup>3</sup>, dentro de ella tenemos una tabla llamada Alumnos y se compone por los siguientes campos:

```
CREATE TABLE `Alumnos` (
  `id` int(11) NOT NULL auto_increment,
  `nombre` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `telefono` varchar(50) NOT NULL,
  `promedio` varchar(10) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM;
```

SQL

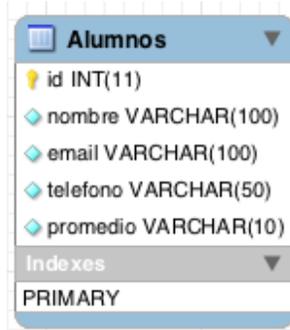


Diagrama ER

Fig. IV.4 Estructura de Base de Datos, Tabla Alumnos

Hasta aquí no hay ningún problema todo es claro, solo estamos definiendo los campos de la tabla Alumnos. Pero ¿A dónde queremos llegar? Bueno vamos a darle un vistazo a lo que el Wizard nos entrega como resultado.

Fig. IV.5 Interfaz para agregar un registro nuevo a la base de datos.

Resultados:	1 al 2 ( total: 2 )	<input type="text"/>	Buscar
Páginas:	1		
Resultados por página:	20	Buscar todos	

CATÁLOGO DE ALUMNOS			
nombre	email	telefono	promedio
Erika	example@gmail.com	01800010053	8.89
Pablo	example@gmail.com	01800010053	8.72

Fig. IV.6 Interfaz de catálogo para la tabla Alumnos

La Fig. IV.5 muestra la interfaz gráfica que nos permite agregar un nuevo registro a la base de datos, específicamente a la tabla Alumnos, pero además se encarga de validar la información de acuerdo al tipo de dato que se requiere, por ejemplo en el campo email no podríamos colocar algo como "http://example.com" ya que se trata de una dirección Web (URL) y no de una dirección de correo, por lo que nos marcaría un error de validación al tratar de agregar esa información.

Con todo esto nos percatamos de otras cosas que van intrínsecas como el área de mensajes de sistema para la validación en este caso, además de una interfaz gráfica usable, útil e intuitiva.

<sup>3</sup> ICO acrónimo de Ingeniería en COmputación

La Fig. IV.6 muestra la interfaz gráfica de catálogo, esta nos sirve para mostrar un listado del contenido de la tabla Alumnos, por lo que nos será de gran ayuda si requerimos visualizar los detalles de un registro, editarlos o borrarlos. Esta interfaz tiene integrados varios módulos del Framework como lo son la paginación, búsqueda y ordenamiento.

**¿Cómo llegamos a este resultado con tan solo la estructura de base de datos y sin escribir una sola línea de código?**

El algoritmo es relativamente sencillo puesto que estamos hablando de automatizar un proceso del cual ya se tiene perfectamente establecido su comportamiento y que es sumamente usado en distintas aplicaciones, lo cual evita que reescribamos algo que ya desarrollamos.

Viendo la Fig. IV.4, IV.5 y IV.6 notaremos que hay algo en común, los campos de la tabla Alumnos se repiten en cada una de las imágenes, desde la estructura de la base de datos hasta el catalogo de Alumnos, lo cual nos ayuda mucho a simplificar este proceso.

Si revisamos en la Fig. IV.4 y la Fig. IV.5 veremos que el nombre de los campos de la tabla Alumnos son reutilizados para la interfaz gráfica de usuario, por lo que estamos ahorrando tiempo en colocar etiquetas y mensajes.

Siguiendo esta misma línea de lógica, estaríamos obviando como se llama cada campo para poder ser validado antes de agregar un registro a la tabla.

Aplicando las reglas de la aritmética pura vamos a citar el siguiente ejemplo:

Sabemos que:

$$2 + 2 = 4$$

y que:

$$\begin{array}{r} 47896 \\ +241259 \\ \hline 289155 \end{array}$$

Por lo que tenemos perfectamente bien establecido el comportamiento del proceso suma, lo que nos lleva a decir que sin importar el valor numérico de los operandos vamos a obtener un resultado basado en un conjunto de reglas que están perfectamente definidas.

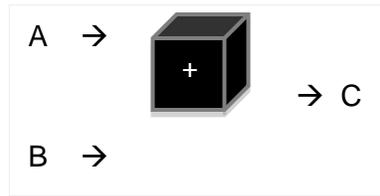
Entonces:

$$A + B = C$$

A y B son nuestros operandos que contienen algún dato numérico y que en base a un proceso llamado suma nos lleva a un resultado C.

En este último ejemplo se usaron literales en vez de valores numéricos, sabemos que cada una de las literales ocuparan un valor determinado en el momento de hacer los cálculos pero mientras describimos de manera general el comportamiento de la suma.

Por lo que si vemos al operador suma como una caja negra, quedaría de la siguiente manera:



Concluimos que A y B son las entradas a nuestro sistema y C el resultado que nos arroja el proceso de la caja negra.

Así de claro también tenemos el proceso de **CRUD**, por que sabemos que estas cuatro operaciones están directamente relacionadas con los campos definidos en la tabla.

Veamos la solución para el caso de “**Create**” de la tabla Alumnos definida en la Fig. IV.4

En lenguaje SQL tendríamos una sentencia como la que sigue:

```
INSERT INTO alumnos (idAlumnos, nombre, email, telefono, promedio) VALUES (.$idAlumnos., '$nombre.', '$email.', '$telefono.', '$promedio.');
```

Donde las literales que inician con signo de pesos (\$) representan al valor que proviene del formulario(datos que el usuario ingreso), tal cual se muestra en la Fig.IV.5

Lo único que puede hacer que los datos en esta sentencia SQL cambien, sería que tuviéramos otra definición de tabla diferente a la presentada en la Fig. IV.4

Vamos a analizarlo de forma integral para resaltar la redundancia tanto en el código como en la interfaz gráfica.

Paso 1.- Definición de tabla (código SQL) :

```
CREATE TABLE `Alumnos` (
  `id` int(11) NOT NULL auto_increment,
  `nombre` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `telefono` varchar(50) NOT NULL,
  `promedio` varchar(10) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM;
```

Paso 2.- Interfaz Gráfica de Usuario (HTML):

Paso 3.-Validación de Datos (código PHP) :

```
$vrfyData->fieldValidate($alumnosData['nombre'],$type);
$vrfyData->fieldValidate($alumnosData['email'],$type);
$vrfyData->fieldValidate($alumnosData['telefono'],$type);
$vrfyData->fieldValidate($alumnosData['promedio'], $type);
```

Paso 4.- Inserción a la base de datos (código PHP / SQL) :

```
INSERT INTO alumnos (idAlumnos, nombre, email, telefono, promedio) VALUES ('.$idAlumnos.', '.$nombre.',
'.$email.', '.$telefono.', '.$promedio.'
```

Esto se convierte en un ciclo, por que a partir de la estructura de la base de datos(paso 1), generamos la interfaz gráfica(paso 2) la cual es trivial, una vez que el usuario interactúa con la interfaz gráfica e ingresa los datos correspondientes se le regresa el control al sistema para que valide los datos(paso 3) en caso de que estos sean correctos se procede a insertarlos(paso 4) en la tabla definida(paso 1)

Resumiendo la solución queda de la siguiente manera:

Paso 1. Definición de la tabla de 1...N campos a guardar.

Paso 2. Generación Automática de un formulario de 1...N campos a llenar.

Paso 3. Validación de 1...N campos antes de guardarlos.

Paso 4. Ejecución de una sentencia SQL de 1...N campos para ser guardados en la tabla.

Estamos hablando de un comportamiento establecido y como principal variable tenemos la longitud de campos, ya que como mínimo debe de haber uno pero no sabemos el limite que se requiera cada vez que sea necesario desarrollar un Backend, si no más bien dependerá de los requerimientos de la aplicación en específico. Esto es sumamente sencillo de resolver con “Ciclos” y “Arreglos” dentro de un lenguaje de programación, en este caso específicamente PHP. Veamos un ejemplo sencillo del Paso 2.

```
1 <?
2
3 for($i=0; strlen($data)-1; $i++){
4
5     $Text = new Text($data['valor']);
6     $inputDataTable->addDataRow($data['campo'], $Text->getBody());
7
8 }
9
10
11 //Prueba de escritorio
12 $data[0]['campo']='nombre';
13 $data[0]['valor']='Pablo';
14
15 ?>
```

Fig IV.7

De la línea 3 a la línea 8 de la Fig. IV.7 observamos un “ciclo” que por medio del Framework nos permite sintetizar la generación de una tabla de N filas y dos columnas, conteniendo en la columna izquierda la etiqueta del dato requerido y en la columna del lado derecho un campo de texto para agregar la información solicitada, de esta manera automatizamos la generación de código HTML para la rápida creación de una interfaz de usuario.

Todo esta en función de la variable \$data, en las líneas 12 y 13 vemos como esa variable adopta dos datos. Haciendo una prueba con esos datos el resultado en pantalla seria el siguiente.

nombre

Para fines prácticos este ejemplo solo tiene una iteración pero queda claro que esto podría generar N filas necesarias para cada uno de los campos requeridos por la tabla en la base de datos. Por lo que con esto nos olvidamos de escribir dos veces algo que ya desarrollamos.

Una vez más con este ejemplo vimos que el Framework nos ayuda a mantener un código limpio y estándar, además de ayudarnos con las tareas comunes en el desarrollo de aplicaciones, evitando con esto los errores de dedo y ahorrándonos el tiempo de debugeo<sup>4</sup> de los mismos.

De forma idéntica, siguiendo la misma lógica aquí presentada se resuelven los otros tres pasos restantes para completar el CRUD. Para fines prácticos para este trabajo nos avocamos al paso de "Create" que sí bien no es nada complejo, si es el más extenso de los cuatro.

## Demostración del Proyecto

Ya hemos descrito teóricamente el cómo se ha desarrollado el proyecto y ahora pasaremos a la demostración del mismo.

Nuestra primera pantalla del Wizard es muy sencilla solo nos da la opción de seleccionar la tabla con la cual queremos trabajar, previamente el Wizard ya se ha conectado a la base de datos para poder mostrar las tablas existentes, y por medio de una lista desplegable seleccionamos la que se requiera.

Una vez seleccionada la tabla, el Wizard se conecta a la base de datos para obtener los campos definidos, esto servirá como entrada a nuestro sistema, para definir cuantos campos existen y cual es el nombre de cada uno, incluso podemos saber de que tipo es.

Esto nos lleva a la segunda pantalla y la principal del Wizard que a continuación se presenta:

WIZARD ADMIN CODE						
Validacion de Campos						
Campo	Validacion	Funcion de Validacion				
idAlumnos:	no					
nombre:	obligatorio	getSpanishStringNumberRegExp(2, 100)				
email:	obligatorio	getEmailRegExp()				
telefono:	obligatorio	getSpanishStringNumberRegExp(2, 50)				
promedio:	obligatorio	getSpanishStringNumberRegExp(2, 10)				
Catalogo						
Campo	Seleccionar	Desplegar	Etiqueta	Buscar	Orden	Foco
idAlumnos:	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>		<input type="radio"/>
nombre:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nombre	<input checked="" type="checkbox"/>	ASC	<input checked="" type="radio"/>
email:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	email	<input checked="" type="checkbox"/>	ASC	<input type="radio"/>
telefono:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	telefono	<input checked="" type="checkbox"/>	ASC	<input type="radio"/>
promedio:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	promedio	<input checked="" type="checkbox"/>	ASC	<input type="radio"/>
<input type="button" value="GenerarCodigo"/>						

Fig. IV.8

<sup>4</sup> Termino usado en el desarrollo de software y que se refiere a la depuración de errores.

En la Fig. IV.8 se muestra una interfaz grafica, que nos permite configurar para cada uno de los campos de la tabla Alumnos definida en la Fig. IV.4, en esta interfaz podremos adecuar los requerimientos de nuestra aplicación.

Esta dividida en dos secciones principales:

- ✓ Validación de Campos:
  - Aquí seleccionamos de acuerdo a la información que se requiere para cada campo, la validación adecuada para mantener la congruencia de los datos en la base de datos, en caso que se requiera podremos evitar la validación.
  
- ✓ Catalogo:
  - Esta sección esta enfocada en configurar nuestro listado de información, de tal manera que sea lo más usable para el usuario final. Podremos mostrar o no cierto campo, agregar algún ordenamiento determinado o quizás cambiar la etiqueta.

Con este sencillo Wizard fundamentado en el Framework es suficiente para generar el código compatible con el mismo Framework y que se puede probar inmediatamente que se le de click al botón “Generar Código”. De esta manera estamos ahorrando varias horas de programador y ahorrándonos el trabajo de algo que ya hemos hecho en otras ocasiones, trayendo consigo todos los beneficios de los que se ha hablado a lo largo de cada uno de los capítulos de este trabajo.

El costo de desarrollar un Wizard de este tipo ya habíamos mencionado que esta directamente relacionado con el dominio que tenga el programador del Framework seleccionado pero que de una forma u otra es un costo relativamente pequeño en comparación con los beneficios que a continuación se mencionan.

El Wizard nos ahorrara codificar alrededor de 394 líneas de código por cada una de las tablas de la base de datos. Suponiendo que una base de datos pequeña tiene alrededor de 10 a 15 tablas, estaríamos hablando que nos ahorra el trabajo de codificar por lo menos 3,940 líneas, lo cual trasladado a la línea del tiempo nos daría un ahorro medido en unidades de hora, que dependiendo de la habilidad del programador y del dominio del Framework podría incluso extenderse a días para cubrir los requerimientos solicitados inicialmente.

Por lo que estamos ahorrando una gran cantidad de tiempo que es muy benéfica para garantizar la entrega de cierto proyecto, pero además obtenemos un software de calidad, que respeta los estándares establecidos internacionalmente conteniendo una interfaz usable e intuitiva.

Algo muy importante de mencionar es que al respetar un estándar será mucho más fácil mantener el código y en caso de ser requerido escalar la aplicación y/o conectarla con otra.

## Conclusiones

A través de este trabajo logramos explorar varias tecnologías de punta que se involucran en el desarrollo de software para Web, lo cual nos deja una gran retroalimentación ya que la próxima vez que se decida realizar algún proyecto de esta índole contaremos con las bases mínimas necesarias para realizarlo, además de contar con referencias de primera mano para consultar cualquier cuestión específica.

Un logro alcanzado y que hay que destacar, sin duda son las “mejores practicas” que se exponen en cada una de las fases que componen el desarrollo de software para Web, desde la selección de la plataforma, pasando por el análisis para concluir en el desarrollo.

No solo considere las “mejores practicas” para el óptimo uso de la tecnología, si no también me permití transmitir a cualquier programador que inicie a desenvolverse profesionalmente; mis propias experiencias, para que sean consideradas y que logren superar los retos que la industria les pondrá en frente.

Esperando que por medio de esta transmisión del conocimiento se pueda incrementar el nivel de programación de una persona y así mismo superar en conjunto las metas que te exige esta profesión, como lo son el contar con una correcta fase de análisis, realizar una clara documentación, desarrollar con base en una metodología, entre muchas otras.

Un consejo que me atrevo a dar a cualquier programador es que no se tiene que desarrollar lo que ya esta hecho, se tiene que construir algo mejor con lo que ya esta realizado, es por eso que es necesario contar con una plataforma que nos sirva de apoyo para lograr los objetivos de manera exitosa.

De manera muy particular me refiero a que hay que contar con una buena colección de herramientas que nos permitan elaborar nuestro trabajo y no solo conformarnos con la pertenencia de las mismas, hay que aprender como usarlas y saber cuando es mejor usar una herramienta que otra, encontrar las bondades y los requerimientos para el óptimo rendimiento.

Llegar a este punto quizás requiera varios años de experiencia, la idea de este trabajo es que el programador cuente con una noción de cómo enfrentar aquellos juegos de destreza que te asignan los jefes e ir más allá de lo que se te requiere colocándole “la cereza al pastel”.

Materializando las ideas que se plantearon desde un inicio llegamos al punto de fabricar un Wizard, en el cual se aplican los conceptos adquiridos en los primeros capítulos, lo más importante para un programador novato ó con varios años de experiencia dentro de este trabajo de titulación es que aprenda el concepto de Metaprogramación y que lo lleve a la practica, ya que el primer beneficiado será el mismo, tanto profesionalmente como personalmente.

Basando el desarrollo de nuevas aplicaciones en código que ya ha sido probado y usado en otras aplicaciones, se incrementa la fiabilidad y se reduce en general el tiempo de fabricación del producto, resumiéndose en un gran beneficio para todos, desde el programador, el equipo de desarrollo, la imagen y prestigio de la empresa fabricante y la buena experiencia del usuario final.

En el capítulo 4, se explica el procedimiento de cómo realizar un Wizard aplicando el concepto de Metaprogramación, con la finalidad de evitar solicitar permisos especiales para colocar código fuente de la aplicación y del Framework utilizado, el código mostrado fue explícitamente para ejemplificar como se realizaría el proyecto.

## Glosario

### Base de datos

Conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente. En una base de datos, la información se organiza en campos y registros. Los datos pueden aparecer en forma de texto, números, gráficos, sonido o vídeo.

### Bug

Término aplicado a los errores descubiertos al ejecutar cualquier programa informático. Fue usado por primera vez en el año 1945 por Grace Murray Hooper, una de las pioneras de la programación moderna, al descubrir cómo un insecto (bug en inglés) había dañado un circuito de la computadora.

### Clase

Estructura que define como son los objetos, indicando sus atributos y sus acciones.

### Código fuente

Conjunto de instrucciones que componen un programa, escrito en cualquier lenguaje. En inglés se dice "**source code**". Hay programas de código abierto y "de código cerrado" como por ejemplo Windows, Photoshop, y la mayoría de los programas comerciales, en donde el código es inaccesible y por lo tanto no se puede alterar la estructura del programa.

### ECMA

Es una organización internacional basada en membresías de estándares para la comunicación y la información. Los objetivos de ECMA International son desarrollar, en cooperación con las organizaciones de estándares nacionales, europeas e internacionales, estándares y reportes técnicos con el fin de facilitar y estandarizar el uso de las Tecnologías de Información y Comunicación y Dispositivos Electrónicos.

### Extensibilidad

En ingeniería de software se usa este concepto para decir que un software prevé minimizar el impacto de cambios futuros en las funciones existentes del sistema. Lo que permite al programador codificar una aplicación específica.

### IEC

Comisión Electrotécnica Internacional

### Instancia

llámese instancia, al objeto creado a partir de una clase.

### ISO

Organización Internacional para la Estandarización

### Modularidad

Es la capacidad que tiene un sistema de ser estudiado, visto ó entendido como la unión de varias partes que interactúan entre sí y que trabajan para alcanzar un objetivo común, realizando cada una de ellas una tarea necesaria para conseguir dicho objetivo.

### Open source / Software Libre

Código fuente abierto o software libre, se refiere a un programa cuyo código fuente está disponible al público general, para usar y modificar. El software libre no es siempre software gratuito.

### URL

Fueron usadas por primera vez por Tim Berners-Lee en 1991, para permitir a los autores de documentos establecer hiperenlaces en la World Wide Web.

**Software**

Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

**SaaS**

Software as a Service (Software como Servicio) es un modelo de distribución de software en donde la compañía de tecnologías de información y comunicación IT provee el servicio de mantenimiento, operación diaria, y soporte del software usado por el cliente. Es tener la información, el procesamiento, los insumos y los resultados de la lógica de negocio del software hospedado en la compañía de IT.

## Bibliografía

Ingeniería de Software

Ian Sommerville

Pearson Educación, S.A., Madrid, 2005

<http://books.google.com/books?id=gQWd49zSut4C&lpq=PA656&ots=s446wnuAsi&dq=define%20instanciar&pg=PA24#v=onepage&q=define%20instanciar&f=false>

El Desarrollo del Framework Orientado al Objeto

M. en C. Marcus Eduardo Markiewicz

<http://www.acm.org/crossroads/espanol/xrds7-4/frameworks.html>

Tecnología Java

<http://java.sun.com/>

Tecnología PHP

<http://www.php.net/>

Microsoft .Net Framework

<http://msdn.microsoft.com/en-us/netframework/>

Java Framework

[http://www.jfwk.com/what\\_is.html](http://www.jfwk.com/what_is.html)

Your guide for developing usable & useful Web sites.

<http://www.usability.gov>

Usability

<http://www.useit.com/alertbox/20030825.html>

W3C – HTML estandars

<http://www.w3.org>

WebKit

Open source web browser engine

<http://www.webkit.org>

Gecko

layout engine developed by the Mozilla Project

<https://developer.mozilla.org/en/Gecko>

JavaScript

<https://developer.mozilla.org/es/JavaScript>

AJAX

<http://www.w3schools.com/Ajax/>

Prototype

<http://www.prototypejs.org/>

jQuery

<http://jquery.com/>

Mootols

<http://mootools.net/>

Script.aculo.us

<http://script.aculo.us/>

Google Web Toolkit

<http://code.google.com/intl/es/webtoolkit/>

Yahoo Yui Library

<http://developer.yahoo.com/yui/>

Adobe Spry

<http://labs.adobe.com/technologies/spry/>

Amazon EC2

<http://aws.amazon.com/ec2/>

Google Apps

<http://www.google.com/apps/>

Microsoft Azure

<http://www.microsoft.com/windowsazure/>

Google Chrome OS

<http://www.chromium.org/chromium-os>

El Universal - Google Chrome OS

<http://www.eluniversal.com.mx/notas/641120.html>

<http://www.eluniversal.com.mx/articulos/56641.html>