

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR  
FACULTAD DE INGENIERÍA  
ESCUELA DE SISTEMAS**

**ELABORACIÓN DEL PORTAL WEB INMOBILIARIO “*VIVIR EN ECUADOR*”**

**CERÓN GRANJA DAVID ANDRÉS**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN SISTEMAS Y COMPUTACIÓN**

**QUITO, 2009**

## **DEDICATORIA**

*Dedicado a mis padres Esmeralda y Fabián, que con tanto cariño  
y esfuerzo me dieron la mejor educación y el mejor ejemplo.*

*A ti mami, por enseñarme que no existen metas imposibles de alcanzar,  
por ayudarme en mis proyectos desde que tengo memoria  
y por seguirme ayudando desde el Cielo.*

*A ti papi, por ser un ejemplo de vida, por salir adelante siempre  
y por enseñarme que la vida hay que vivirla intensamente.*

## **AGRADECIMIENTO**

*Agradezco a Dios por fortalecer mi corazón e iluminar mi mente  
y por darme la oportunidad de alcanzar este logro.*

*A mis padres, Esmeralda y Fabián, por su dedicación y apoyo incondicional.*

*A mis hermanos Sofía y Fabián, y a toda mi familia por darme ánimo  
y por brindarme su cariño.*

*A mis amigos por confiar y creer en mí  
y sobre todo por su valiosa amistad.*

*Y a mis profesores, quienes con paciencia supieron  
compartir invaluable conocimientos.*

## ÍNDICE DE CONTENIDOS

INTRODUCCIÓN.....	1
<b>CAPÍTULO I</b>	
PORTALES WEB.....	3
1.1 Evolución de los portales Web.....	3
1.2 Características de los portales Web.....	5
1.3 El concepto Web 2.0.....	6
<b>CAPÍTULO II</b>	
APLICACIONES WEB CON BASES DE DATOS.....	7
2.1 Software Libre.....	8
2.2 El lenguaje de programación <i>PHP</i> .....	11
2.3 El sistema de gestión de bases de datos <i>MySQL</i> .....	12
2.4 <i>PHP</i> y <i>MySQL</i> : ¿la combinación perfecta?.....	13
<b>CAPÍTULO III</b>	
METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	16
3.1 Metodologías estructuradas.....	19
3.2 Metodologías orientadas a objetos.....	20
3.3 Metodología para desarrollo Web.....	20
<b>CAPÍTULO IV</b>	
ELABORACIÓN DEL PORTAL INMOBILIARIO	
“ <i>VIVIR EN ECUADOR</i> ”.....	23
4.1 Fase de preparación.....	23
4.1.1 Entrevista con el cliente para acordar especificaciones del proyecto.....	23
4.1.2 Analizar el alcance del proyecto.....	24
4.1.3 Elaborar el cronograma de trabajo y el presupuesto.....	28

4.2	Fase de diseño.....	30
4.2.1	Entrevista con el cliente para definir el diseño.....	30
4.2.2	Presentar propuestas de diseño.....	31
4.2.3	Corregir o modificar el diseño.....	32
4.3	Fase de desarrollo.....	33
4.3.1	Elaborar la página de inicio ( <i>homepage</i> ).....	34
4.3.2	Iniciar la programación del sitio Web.....	34
4.3.3	Solicitar los contenidos para las páginas del proyecto.....	35
4.3.4	Publicar el avance del proyecto en el servidor Web.....	36
4.3.5	Incorporar correcciones o modificaciones.....	36
4.4	Fase de finalización.....	37
4.4.1	Revisión final de contenidos.....	37
4.4.2	Cargar el proyecto en el servidor Web.....	37
4.4.3	Efectuar correcciones.....	38
4.4.4	Aprobación final del cliente.....	38

## **CAPÍTULO V**

	CONCLUSIONES Y RECOMENDACIONES.....	39
5.1	Conclusiones.....	39
5.2	Recomendaciones.....	40
	BIBLIOGRAFÍA.....	41
	ANEXOS.....	45

## INTRODUCCIÓN

El servicio de *Internet* en Ecuador ha experimentado cambios positivos gracias a la reducción de las tarifas y al incremento de la velocidad de transferencia. Cada día son más las personas que usan *Internet*, no sólo para buscar información sino también para comprar u ofertar bienes y servicios.

En la actualidad, las empresas comienzan a darse cuenta de que estar en *Internet* ha dejado de ser un lujo y se ha convertido en una verdadera necesidad, muchas de ellas lo han hecho casi obligadas, ya sea por sus proveedores o por sus clientes; y al hacerlo han observado cómo el introducir sus empresas en *Internet* mejora significativamente su imagen corporativa.

La elaboración de páginas Web en nuestro país se ha incrementado para satisfacer la creciente demanda. Existe un gran número de empresas y de profesionales que prestan este servicio. Hay también una amplia gama de costos y calidad del mismo.

Se debe tomar en cuenta que un sitio Web es, a menudo, la primera impresión que proyecta una empresa. Una página mal diseñada y visualmente débil puede alejar a los visitantes, dejándolos con una impresión negativa, difícil de corregir. Para evitarlo, se deben incorporar estándares tanto de diseño, como de programación para que una hoja Web sea atractiva, fácil de navegar y guarde la integridad de la información almacenada.

La constructora CAPESSO requiere promocionar sus proyectos de vivienda para incrementar sus ventas y obtener un retorno de sus inversiones de forma más rápida. En la búsqueda por encontrar una alternativa que les permita publicitar sus productos, surgió la idea de realizar un sitio Web que, además de constituirse en una vitrina para sus proyectos inmobiliarios, les proporcione una plataforma para atraer a potenciales clientes.

Entre alternativas como la publicación de anuncios en revistas especializadas, la promoción en medios masivos y la elaboración de un portal Web, los directivos de la

empresa escogieron esta última, ya que el costo de implementación es menor, comparado con las demás opciones.

El portal ofrecerá a los constructores una herramienta para poner sus proyectos inmobiliarios a disposición de potenciales compradores, mientras que a los usuarios se les brindará información de la oferta inmobiliaria, tanto de venta, como de arriendo.

Como herramientas para el desarrollo del portal Web se escogieron el lenguaje de programación *PHP* y el manejador de base de datos *MySQL*, los dos son excelentes exponentes del concepto de software libre. Además, ambas herramientas pueden ejecutarse en servidores *Linux*, lo que reduce significativamente el costo del proyecto, pues así mismo, el costo de licencias y de mantenimiento es bajo.

El presente trabajo iniciará con una reseña de cómo las páginas Web han evolucionado, desde albergar información estática hasta convertirse en portales Web que interactúan con el usuario. Se busca identificar las principales características de un portal Web y abordar el concepto de la *Web 2.0*.

Para la elaboración del portal Web se emplearán herramientas de software libre, por lo que se realizará un breve análisis de las ventajas que ofrece este tipo de software. Enfocándose en el lenguaje de programación *PHP*, el manejador de base de datos *MySQL* y su uso combinado para programar sitios Web.

Se utilizará una adaptación de la metodología propuesta por la empresa Greynium, especializada en la creación de portales Web, profundizando en la metodología, ya que sirve como guía para proyectos orientados a desarrollo Web, los cuales poseen características muy particulares como un tiempo de desarrollo corto y la máxima optimización de los recursos.

## **CAPÍTULO I**

### **PORTALES WEB**

*Internet*, por su naturaleza, carece de una estructura definida, por lo que desde sus inicios se realizaron esfuerzos para facilitar la navegación, dado el creciente número de páginas Web. Fue así como surgieron los buscadores, intentando clasificar la información disponible en la red, pero aún los motores de búsqueda más sofisticados han sido incapaces de registrar todo su contenido. Según *Google*, ellos realizan varias veces al día “una labor computacional equivalente a la de revisar cada una de las intersecciones en cada una las vías de los Estados Unidos, pero en un lugar 50.000 veces más grande y con 50.000 veces más intersecciones”<sup>1</sup>, para que cuando un usuario realice una búsqueda, puedan presentarle un resultado que le satisfaga.

Aunque los buscadores pueden incluir en sus índices de búsqueda a millones de páginas Web, en muchas ocasiones el usuario no sabe qué buscar. Necesita una página que le presente información acorde a sus necesidades e intereses, es aquí donde aparecen los portales Web, destinados a ofrecer al usuario un contenido organizado, que satisfaga la mayoría de sus requerimientos.

#### **1.1 Evolución de los portales Web**

“A finales de 1990 tuvo lugar una revolución que cambiaría nuestras vidas, esta revolución fue el nacimiento de la World Wide Web”<sup>2</sup>. Dos científicos del CERN (Laboratorio Europeo de Física de Partículas), el físico Tim Berners-Lee y el ingeniero en sistemas Robert Cailliau trabajaban en un proyecto cuyo objetivo era transferir información por Internet, utilizando hipertexto, para compartir los datos almacenados en las computadoras del laboratorio. Así nació el primer servidor Web. Para noviembre de 1992 ya existían 26 servidores en el mundo y para octubre de 1993 su número había aumentado a 200.

---

<sup>1</sup> Official Google Blog. *We knew the Web was big...* Internet. <<http://googleblog.blogspot.com/2008/07/we-knew-Web-was-big.html>> Acceso: 08/08/2009.

<sup>2</sup> CERN. *The Website of the world's first-ever Web server.* Internet. <<http://info.cern.ch/>> Acceso: 08/08/2009.

Los primeros usuarios de *La Web* fueron científicos, por lo que las primeras páginas Web tenían como única finalidad el compartir información y consistían en texto e hipervínculos o *links*. La parte gráfica no era relevante y no había interés en incorporar animaciones, audio o video.

Esta situación cambió en 1996, cuando los departamentos de marketing empezaron a interesarse en el enorme potencial comercial que tenía *Internet* y fue *Pepsi* la empresa que lanzó uno de los primeros sitios Web corporativos, intentando crear un mundo virtual para sus visitantes. “El sitio Web incluía arte digital, animaciones en sus logos, avances de películas y las canciones más populares del momento. Esto cambiaría la forma en cómo se diseñaban páginas Web”<sup>3</sup>.

Pero las innovaciones en el diseño trajeron consigo un problema: ¿cómo evitar que cada programador utilice distintos procedimientos para crear una página Web? Fue entonces cuando los estándares Web se hicieron necesarios. El contar con una plataforma común para poder elaborar páginas Web contribuyó a que esta tarea no se convirtiera en algo caótico y a que *Internet* se vuelva más amigable para los usuarios.

A medida que *La Web* siguió ampliando sus fronteras, fue cambiando su enfoque hacia lo que se conoce como una arquitectura orientada a servicios. Los sitios Web dejaron de ser lugares con información y se convirtieron en portales con un conjunto de servicios acoplados. Cada vez que un usuario visita un portal, se siente como en casa, el portal sabe cuáles son sus gustos e intereses. Si visita una tienda *on-line*, no sólo puede efectuar compras sino que el portal conoce su nombre, le recomienda productos basándose en el perfil del usuario, le informa acerca de nuevos productos y le permite valorar los productos que ya compró, para que otros usuarios decidan si vale la pena o no adquirirlos.

---

<sup>3</sup> Richard MacManus. *The Evolution of Corporate Web Sites*. Internet. [http://www.digital-Web.com/articles/the\\_evolution\\_of\\_corporate\\_Web\\_sites/](http://www.digital-Web.com/articles/the_evolution_of_corporate_Web_sites/) Acceso: 08/08/2009.

## 1.2 Características de los portales Web

Un portal Web se define como un punto de partida para el usuario, donde puede encontrar contenido organizado y donde se concentran productos y servicios, orientados ya sea a usuarios en general o a usuarios específicos.

De aquí se desprenden tres tipos de portales:

- Horizontales: son de uso masivo y para público en general.
- Verticales: están dirigidos a ofrecer contenido sobre un tema específico, como un portal de música, empleo, inmobiliario, finanzas o deportes.
- Diagonales: son una mezcla entre el portal horizontal y el vertical, utilizan redes sociales complementadas con contenidos dirigidos a un público muy concreto.

Un portal inmobiliario es un portal vertical, ya que provee información y servicios a un segmento en particular. Con la propuesta de esta disertación se busca dar a conocer los proyectos inmobiliarios de una constructora y además permitir a usuarios registrados ofertar bienes inmuebles.

El objetivo de un portal es fidelizar a sus visitantes, recibir el mayor número de visitas, generando así tráfico hacia su sitio Web y asegurando su permanencia en *Internet*. Un portal debe proveer un ambiente amigable, tanto para usuarios registrados, como para visitantes ocasionales.

En general, los portales Web tienen las siguientes características:

- Permitir a los usuarios publicar y modificar información, a través de un gestor de contenido.
- Proveer una plataforma que combine fácil uso con tecnologías que permitan al usuario intercambiar información con el portal.

Los portales Web son recomendables para proyectos que requieran modificar sus contenidos a menudo, pues su gestor de contenidos hace posible que cualquier cambio sea rápido y sencillo, además de no requerir software adicional, sino que los cambios son hechos desde la misma página Web.

### 1.3 El concepto Web 2.0

El término *Web 2.0* fue creado por Tim O'Reilly en 2004, para referirse un conjunto de principios y prácticas en el desarrollo Web. Esta generación se basa en comunidades de usuarios y en una gama especial de servicios, como las redes sociales, los *blogs* o los *wikis*, que fomentan la colaboración y el intercambio ágil de información entre los usuarios de una comunidad o red social. La *Web 2.0* es también conocida como Web social por el enfoque colaborativo y de construcción social que posee.

El concepto *Web 1.0* corresponde a páginas estáticas *HTML* que no eran actualizadas frecuentemente. La *Web 1.5* se trata de los CMS (*Content Management System*), los cuales generan páginas *HTML* dinámicas creadas desde una base de datos actualizada.

El uso de la *Web 2.0* está orientado a la interacción y a las redes sociales, que actúan más como puntos de encuentro, o *webs* dependientes de usuarios, que como *webs* tradicionales.

La *Web 2.0* no sólo utiliza una gran cantidad de código para que el usuario sienta que intercambia información con una página, sino que además lleva al límite a los servidores para proveer seguridad, rendimiento y escalabilidad. Muchos de los sitios Web que implementan esta tecnología le permiten al usuario personalizar una página de inicio, donde él decide qué elementos la componen: noticias de último minuto, pronósticos de tiempo, calendarios de actividades, reportes de tráfico, índices de la bolsa de valores o juegos. Incluso, algunos le permiten añadir listas de canciones o videos y revisar su correo electrónico. Todo en un mismo lugar y con la apariencia que el usuario defina. “El usuario ya no tiene que buscar la información que le interesa, la información viene hacia él.”<sup>4</sup>

---

<sup>4</sup> Zabir, Omar. *Building a Web 2.0 Portal with ASP.NET 3.5*. Sebastopol, O'Reilly Media, Inc., 2008. Página 4.

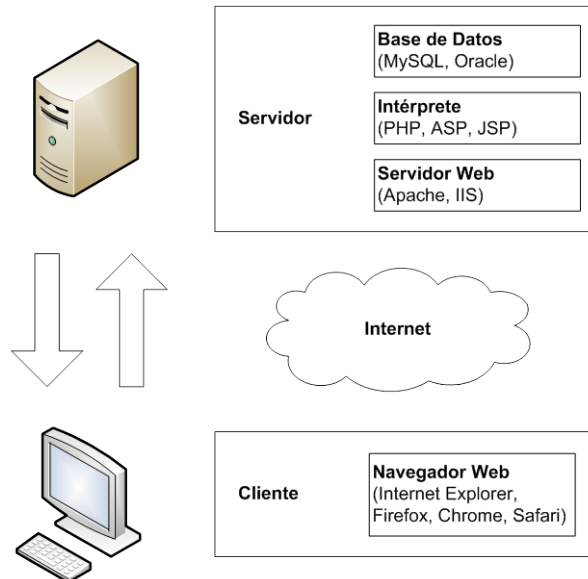
## CAPÍTULO II

### APLICACIONES WEB CON BASES DE DATOS

Para que una página Web deje de ser estática y brinde una mejor experiencia al usuario fue necesario que las bases de datos se incorporaran al desarrollo Web. En un inicio, gracias a *cookies*<sup>5</sup> y a *JavaScript*<sup>6</sup>, había participación del usuario, aunque a un nivel bajo. Este funcionaba siempre y cuando el usuario visitara la página desde un mismo computador, si lo hacía desde otro sus datos ya no podían ser desplegados.

La Web posee una arquitectura cliente-servidor, es decir, necesita un servidor que procese información y se la devuelva a un computador cliente. En la actualidad, los clientes no sólo son computadoras, también se incluyen dispositivos móviles.

La arquitectura con la que funcionan las aplicaciones Web con bases de datos se componen de: cliente, servidor, intérprete y base de datos.



**Figura 2.1. Arquitectura de una aplicación Web**

Fuente: Elaboración propia, basado en Figura I-1 de Greenspan, Jay. *MySQL/PHP Database Applications*. Foster City, IDG Books Worldwide, Inc., 2001.

<sup>5</sup> *Cookie*: es un pequeño archivo que un sitio Web almacena en el disco duro de la persona que lo visita. Tomado de *Vocabulario de ordenadores e Internet* de José Antonio Millán <[http://jamillan.com/v\\_cookie.htm](http://jamillan.com/v_cookie.htm)> Acceso: 12/08/2009.

<sup>6</sup> *JavaScript*: lenguaje de programación interpretado, es decir, que el código no requiere ser compilado, por lo que se lo utiliza en páginas Web. Tomado de EXCEPTION *Diccionario Informático* <<http://www.alegsa.com.ar/Dic/javascript.php>> Acceso: 12/08/2009.

A continuación se explica en qué consiste cada uno de los elementos mencionados anteriormente:

- *Cliente*: las aplicaciones Web usan como cliente al navegador Web, este provee el ambiente necesario para que la aplicación pueda ejecutarse.
- *Servidor*: es el encargado de mantener la comunicación con el navegador, se encuentra a la espera de peticiones del cliente.
- *Intérprete*: son lenguajes de programación que intercambian información con otros programas para indicarle al servidor, exactamente, qué información debe proporcionarle al cliente.
- *Base de datos*: se encarga de almacenar la información que la aplicación requiere.

Para que una aplicación Web sea eficiente se requiere de un lenguaje de programación y un gestor de bases de datos, que sean capaces de comunicarse eficientemente entre sí para lograr aplicaciones robustas, seguras y con tiempos de respuesta cortos. Las herramientas elegidas para el desarrollo de este trabajo fueron: el lenguaje de programación *PHP* y el sistema de gestión de bases de datos *MySQL*, ambas pertenecen a una filosofía denominada software libre, gracias a la cual han podido convertirse en las opciones más utilizadas para la elaboración de aplicaciones Web.

## 2.1 Software Libre

Según la *Free Software Foundation*, el software libre se refiere a la libertad que poseen los usuarios para ejecutarlo, copiarlo, distribuirlo, estudiarlo, cambiarlo o mejorarlo. Se puede catalogar como software libre a todo programa que garantice las siguientes libertades:

- *Libertad 0*: libertad de usar el programa con cualquier propósito.
- *Libertad 1*: libertad de estudiar el funcionamiento del programa y adaptarlo a necesidades particulares.
- *Libertad 2*: libertad de distribuir copias, con lo cual se puede ayudar a otros.
- *Libertad 3*: libertad para mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie.

El software libre no necesariamente debe ser gratuito y, mientras cumpla con las cuatro libertades, puede distribuirse comercialmente. Existe polémica en torno al software libre, debido a la ambigüedad del término en inglés *free software*, que puede interpretarse como software gratuito.

En la década de 1960, el software no era considerado un producto, sino un valor agregado que los vendedores de las grandes computadoras (*mainframes*) proporcionaban a sus clientes para que éstos pudieran usarlos. Era común que entre programadores y desarrolladores de software compartieran libremente sus programas.

Esto cambió en 1980, cuando las compañías impusieron restricciones, a través del uso de acuerdos de licencia, que forzaban a los usuarios a aceptar condiciones que impedían realizar modificaciones al software. En caso de que algún usuario o programador encontrara algún error en la aplicación, lo único que podía hacer era notificar a la empresa desarrolladora para que ésta lo solucionara. Aunque el programador estuviese capacitado para resolver el problema y lo deseara hacer, incluso sin recibir una remuneración por ello, el contrato le impedía que mejorara el software.

Richard Stallman, fundador de la *Free Software Foundation*, cuenta que en el laboratorio donde trabajaba recibieron una impresora como donación. El dispositivo, que era utilizado en red por todos los trabajadores, parecía no funcionar correctamente y cada cierto tiempo el papel se atascaba. Como agravante, no se generaba ningún aviso que se enviara por red e informara a los usuarios de la situación.

La pérdida de tiempo era constante, los trabajadores enviaban a imprimir sus trabajos y al ir a buscarlos se encontraban con la impresora atascada y una cola enorme de trabajos pendientes. Richard Stallman decidió arreglar el problema e implementar el envío de un aviso por red cuando la impresora se bloqueara. Para ello, necesitaba tener acceso al código fuente de los controladores de la impresora, por lo que pidió a la empresa propietaria la información que necesitaba, indicándoles qué era lo que pretendía realizar, pero la empresa se negó a entregarle el código fuente.

Con este antecedente, en 1984, Richard Stallman comenzó a trabajar en el proyecto *GNU*<sup>7</sup> y un año más tarde instituyó la *Free Software Foundation*. Stallman introdujo la definición de software libre y el concepto de “copyleft”<sup>8</sup>, que desarrolló para otorgar libertad a los usuarios y para restringir las posibilidades de apropiación del software.

El movimiento del software libre hace énfasis en los aspectos morales y éticos del software, viendo a la excelencia técnica como un producto secundario. El movimiento de código abierto (*open source*) ve a la excelencia técnica como el objetivo prioritario, siendo el compartir el código fuente un medio para dicho fin. El término código abierto se asocia a una filosofía, a unos valores y a un criterio diferente por el cual las licencias son aceptables. “El movimiento por el software libre y el movimiento por el código abierto hoy son movimientos separados con visiones y metas diferentes, aunque trabajemos juntos en algunos proyectos prácticos”<sup>9</sup>.

Un proyecto de software libre tiene un desarrollo más rápido, comparado con un software comercial, ya que son miles de usuarios los que modifican y mejoran el código continuamente, en lugar de esperar que la empresa desarrolladora lance una versión mejorada del producto.

El contar con el respaldo de toda una comunidad de usuarios hace mucho más fácil la implementación de soluciones y evita lo que en inglés se denomina “*reinventar la rueda*” (*Reinvent the wheel*). En muchas ocasiones, un programador tiene que reescribir cientos de líneas de código que otro usuario ya ejecutó y probó. La ventaja del software libre es que un programador puede partir del trabajo de otro para mejorar lo que ya está hecho, en lugar de reescribirlo todo, ahorrando así valioso tiempo para el desarrollo del proyecto.

---

<sup>7</sup> Se inició en 1984 con el objetivo de crear un sistema operativo completo tipo Unix de software libre. Tomado de <<http://www.gnu.org/home.es.html>> Acceso: 12/08/2009.

<sup>8</sup> Copyleft ayuda a un programador que quiere contribuir mejoras al software libre obteniendo permiso para hacerlo. Tomado de <<http://www.gnu.org/copyleft/copyleft.es.html>> Acceso: 12/08/2009.

<sup>9</sup> Richard Stallman. *Por qué «software libre» es mejor que «código abierto»*. Internet. <<http://www.gnu.org/philosophy/free-software-for-freedom.es.html>> Acceso: 12/08/2009.

Hay que mencionar también que los costos al utilizar herramientas de software libre son bajos. Para el caso de un portal Web el alojamiento en un servidor *Linux*, comparado con el de un servidor *Windows*, es menor debido a que los costos de licencias y de mantenimiento son bajos.

## 2.2 El lenguaje de programación *PHP*

*PHP* es un lenguaje de programación diseñado originalmente para la creación de páginas Web dinámicas. Se usa principalmente para interpretación del lado del servidor (*server-side scripting*), le indica qué información debe procesar para entregarle al cliente.

*PHP* es un acrónimo recursivo que significa: *Hypertext Pre-processor*, que inicialmente se denominó *Personal Home Page Tools* y fue creado por Rasmus Lerdorf en 1994. Está publicado bajo *PHP License* y la *Free Software Foundation* lo considera un software libre.

Su popularidad va en aumento y en “abril de 2007, casi 21 millones de dominios lo utilizaban, según una investigación realizada por *Netcraft*”<sup>10</sup>. Este lenguaje facilita el diseño Web y la programación, gracias a las múltiples funcionalidades que posee. Utiliza una sintaxis similar a la del lenguaje C pero simplificada, por lo que muchos usuarios se sienten cómodos programando en él.

Algunas de las principales ventajas de *PHP* son:

- Rapidez: debido a que se encuentra integrado en el código *HTML*.
- Fácil de usar: *PHP* contiene funciones predefinidas para elaborar páginas Web dinámicas.
- Independiente del sistema operativo: puede ser ejecutado bajo *Windows*, *Linux* y *Mac OS*.
- Soporte técnico: una comunidad de usuarios provee soporte a través de foros y listas de correo electrónico.
- Seguridad: el usuario no puede ver el código, éste se ejecuta directamente en el servidor.
- Soporte para bases de datos: fue diseñado con la funcionalidad de comunicarse con bases de datos y liberar al programador de detalles técnicos para conectarse a ellas.

---

<sup>10</sup> *Netcraft. Usage Stats for April 2007*. Internet. <<http://www.php.net/usage.php>> Acceso: 15/08/2009.

- Configurable: la licencia de software libre permite a los programadores modificar el código fuente de *PHP* para adaptarse a situaciones específicas.

“Su característica más sobresaliente es la habilidad que posee para interactuar con bases de datos. No se necesitan conocer detalles técnicos para conectarse a una base de datos e intercambiar información con ella”<sup>11</sup>. Basta con indicarle dónde está almacenada la base de datos, cuál es su nombre, un usuario y una contraseña y *PHP* se encarga del resto. Se conecta con la base de datos, procesa las instrucciones y retorna la información solicitada. *PHP* trabaja en conjunto con el servidor Web y aún cuando soporta algunos servidores Web, trabaja mejor con *Apache*, que también es software libre y actualmente “gestiona el 51% de todos los sitios Web”<sup>12</sup>.

### 2.3 El sistema de gestión de bases de datos *MySQL*

*MySQL* es un sistema de gestión de bases de datos relacionales rápido y fácil de usar. Fue diseñado para ofrecer alta velocidad y para ello tuvieron que sacrificar algunas funciones. A pesar de contar con menos funcionalidades que sus competidores, *MySQL* cuenta con las funciones necesarias para implementar aplicaciones con bases de datos.

*MySQL* es distribuido bajo dos tipos de licencia: la *GNU GPL*, que es gratuita, y una comercial, a un costo moderado. Fue desarrollado por una empresa sueca llamada *MySQL AB* en 1995, que en enero de 1998 fue adquirida por *Sun Microsystems* y ésta a su vez fue adquirida por *Oracle Corporation*, en abril de 2009.

Algunas de las principales ventajas de *MySQL* son:

- Rapidez: uno de los objetivos de los desarrolladores, cuando crearon *MySQL*, fue darle velocidad, aún sacrificando otras funcionalidades.
- Bajo costo: es gratuito bajo licencia *GPL* y aún bajo licencia comercial tiene un costo razonable.

---

<sup>11</sup> Valade, Janet. *PHP & MySQL For Dummies*. Indianapolis, Wiley Publishing, Inc., 2da edición, 2004. Página 15.

<sup>12</sup> *Netcraft. August 2009 Web Server Survey*. Internet. < [http://news.netcraft.com/archives/2009/08/31/august\\_2009\\_Web\\_server\\_survey.html](http://news.netcraft.com/archives/2009/08/31/august_2009_Web_server_survey.html) > Acceso: 15/08/2009.

- Independiente del sistema operativo: puede ser ejecutado en una amplia gama de sistemas operativos, incluyendo *Windows*, *Linux*, *Solaris*, *FreeBSD*, *OS/2*, *Mac OS*, entre otros.
- Soporte técnico: una gran cantidad de usuarios provee soporte a través de foros y listas de correo electrónico. Además, la empresa desarrolladora ofrece soporte especializado, a un precio cómodo.
- Seguridad: cuenta con un sistema flexible de autorización, que permite asignar privilegios sobre la base de datos a usuarios específicos o a grupos de usuarios, además utiliza contraseñas encriptadas.
- Soporte para bases de datos grandes: soporta bases de datos de hasta 50 millones de registros. El tamaño predeterminado para una tabla es de 4 *gigabytes*<sup>13</sup> pero puede soportar tablas de hasta 8 millones de *terabytes*<sup>14</sup>.
- Configurable: la licencia de software libre permite a los programadores modificar el código fuente de *MySQL* para adaptarse a sus necesidades.

El sistema de base de datos *MySQL* se compone de un servidor *MySQL* y de algunas aplicaciones que le asisten. El servidor es el administrador de todo el sistema y está a la espera de establecer conexiones para después procesar las instrucciones que recibe del cliente. Por ejemplo, “cuando el usuario le envía un mensaje al servidor *MySQL* pidiéndole que cree una nueva base de datos, este se encarga de realizar todas la tareas relacionadas con la orden que recibe: crea un nuevo subdirectorio y los archivos necesarios para que la nueva base de datos tenga una estructura y un formato acordes a las necesidades del usuario”<sup>15</sup>.

## 2.4 *PHP* y *MySQL*: ¿la combinación perfecta?

*PHP* incluyó soporte para *MySQL* desde su tercera versión y gradualmente lo ha ido mejorando. En su versión actual (5.3.1), *PHP* ha logrado formar un vínculo muy estrecho con *MySQL*. “Esta pareja se ha beneficiado del gran esfuerzo que ha hecho la comunidad

---

<sup>13</sup> 1 *gigabyte* equivale a 1024 *megabytes*.

<sup>14</sup> 1 *terabyte* equivale a 1024<sup>2</sup> *megabytes*.

<sup>15</sup> *Ibid.* 11, Página 14.

de desarrolladores para que cuando trabajen juntos lo hagan en armonía y sin problemas”<sup>16</sup>.

Adicionalmente, ambas herramientas son software libre, por lo que los desarrolladores pueden ahorrarles a sus clientes mucho dinero en cuestión de licencias. Las aplicaciones que han utilizado la combinación *PHP-MySQL* abarcan: gestores de contenido, buscadores y sistemas de reportes en línea. En esencia, cualquier aplicación que utilice una base de datos y un navegador puede desarrollarse utilizando esta combinación.

El ambiente típico en el cual se desarrollan las aplicaciones Web que utilizan *PHP* y *MySQL* se denomina *LAMP* y proviene de *Linux*, *Apache*, *MySQL* y *PHP*, y constituye una plataforma de desarrollo, donde cada elemento cumple una función importante:

- *Linux*: provee el sistema operativo al servidor.
- *Apache*: es el servidor Web e interpreta las peticiones *HTTP* y se las comunica al intérprete.
- *MySQL*: almacena la información, acepta conexiones del intérprete y manipula la información.
- *PHP*: interpreta y ejecuta las sentencias de código y devuelve el resultado al servidor Web.

Para programar el portal Web que se propone en este trabajo se utilizará *PHP*, que es lo suficientemente flexible como para realizar tareas sencillas, como desplegar variables y tareas más complejas, como verificar que el usuario haya ingresado su contraseña de forma correcta. La tarea más común que realizará es la de mover información desde y hacia la base de datos y, gracias a sus funciones integradas, será una tarea que no demandará mucho esfuerzo.

Cuando se necesita ingresar o extraer información de la base de datos, se utilizan sentencias específicas para establecer una conexión y darle instrucciones al servidor de bases de datos. *MySQL* responde con un mensaje indicando si la operación fue exitosa o si existió algún error en el proceso. Si la instrucción fue extraer información de la base de

---

<sup>16</sup> Vaswani, Vikram. *How to Do Everything with PHP & MySQL*. New York, McGraw-Hill, 2005. Página 17.

datos, *MySQL* proporcionará la información solicitada y *PHP* la almacenará en una ubicación temporal para que esté disponible, ya sea para manipularla o desplegarla.

Como un gestor de bases de datos relacionales, *MySQL* puede almacenar información realmente compleja, mientras que *PHP*, como un lenguaje de programación interpretado, puede manipular la información, ya sea antes de ser almacenada en la base de datos o antes de ser presentada al usuario. Juntos, *PHP* y *MySQL*, pueden ser utilizados para crear una aplicación Web con una base de datos muy sofisticada y estable.

## CAPÍTULO III

### METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Una metodología de desarrollo de software es un conjunto de procedimientos, técnicas, herramientas y documentos que se emplean para elaborar un nuevo software.

En una metodología se indican, paso a paso, las actividades que se deben realizar para lograr el producto informático deseado y las personas que participan en el desarrollo de esas actividades. Además, se detalla la información necesaria para comenzar una actividad y el resultado que se obtiene de la misma.

La diferencia entre metodología y modelo de ciclo de vida de un software es que, mientras este último indica cuáles son los productos que obtendremos a lo largo de la elaboración, la metodología establece cómo obtenerlos.

Las metodologías de desarrollo se volvieron necesarias para ayudar al Departamento de Defensa de los Estados Unidos, pues tenía muchos problemas con el software que encargaba desarrollar a otras empresas: los presupuestos se disparaban y las fechas de entrega se postergaban. En 1983 se conformó un comité de expertos, el cual concluyó que era necesario crear un instituto de ingeniería del software. La Universidad Carnegie Mellon de Pittsburgh ganó el concurso en 1985, creándose así el SEI (*Software Engineering Institute*), que desarrolló y mantiene el *CMM (Capability Maturity Model)*.

El *CMM* provee a las organizaciones los elementos necesarios para generar procesos eficaces, que ayudan a mejorar su rendimiento y sus procesos. “Ayuda a integrar funciones típicamente separadas, establecer metas y prioridades para la mejora de procesos, proporciona una guía para procesos de calidad y proporciona un punto de la referencia para valorar procesos actuales”<sup>17</sup>.

---

<sup>17</sup> Software Engineering Institute. *Capability Maturity Model Integration*. Internet. < <http://www.sei.cmu.edu/cmml/> > Acceso: 18/08/2009.

Inicialmente fue creado para evaluar y mejorar la capacidad de los contratistas de software del Departamento de Defensa de los Estados Unidos, pero a través de los años el modelo *CMM* se convirtió, a nivel mundial, en el más alto estándar de ingeniería para todo tipo de compañías. Está fundamentado en prácticas reales de éstas y en los mejores procesos de desarrollo de software.

El *CMM* está compuesto por 316 prácticas, agrupadas en 18 áreas y distribuidas en una jerarquía de cinco niveles, a través de las cuales una organización alcanza progresivamente mayor calidad, productividad y menores costos en el desarrollo de software. Los niveles van desde el 1, que representa el estado caótico, hasta el 5, que representa el estado de optimización continua. El *CMMI* (*Capability Maturity Model Integration*) es un modelo posterior que fue elaborado en el año 2002 y sus siglas en inglés significan Modelo de Madurez de Capacidad Integrado.

Los niveles que componen el modelo *CMMI* son:

- *Nivel 1 o inicial*: en este nivel se encuentran todas las empresas que no tienen procesos. Los presupuestos se disparan, no es posible cumplir con los plazos de tiempo y no hay control sobre el estado del proyecto.
- *Nivel 2 o repetible*: el éxito de los resultados obtenidos debe poder repetirse. El proyecto es controlado durante el desarrollo del mismo y es posible conocer su estado en todo momento.
- *Nivel 3 o definido*: la forma de desarrollar proyectos está definida. Esto que decir que está documentada y existen métricas para alcanzar objetivos concretos.
- *Nivel 4 o cuantitativamente gestionado*: los proyectos usan objetivos medibles para satisfacer las necesidades de los clientes y de la organización.
- *Nivel 5 u optimizado*: los procesos están orientados a la mejora de las actividades, que además, mediante el uso de métricas son identificadas, evaluadas y puestas en práctica.

Hace algunos años, cada programador tenía su propia forma de elaborar una aplicación y era una tarea que podía realizarla sin necesidad de más colaboradores. “El programador era una especie de mago que hacía su aplicación y sólo él sabía cómo funcionaba”<sup>18</sup>. Pero con el tiempo, las aplicaciones se volvieron cada vez más grandes y complicadas. Se requerían varios desarrolladores y plazos de entrega más amplios para terminar una aplicación. Actualmente, el elemento más importante en una empresa desarrolladora de software es disponer de profesionales altamente calificados. Sin embargo, esto no asegura el éxito de un proyecto, pues existe el riesgo de que no haya un auténtico trabajo en equipo, debido a la forma de desarrollar software de cada individuo. En el desarrollo de software, el objetivo es establecer un entorno que no sólo mejore la productividad del programador, sino que también ayude a la creación de mejores productos.

El modelo *CMMI* nos indica que aún los mejores informáticos necesitan un entorno disciplinado y estructurado, en el cual puedan realizar un trabajo en equipo para lograr productos con alta calidad. Un programador debe ser consciente de que lo que realiza es un componente que se combinará con otros para formar un sistema y que este componente debe poseer los principios de la ingeniería de software para que el sistema final sea satisfactorio.

Si bien es cierto, el uso de una metodología implica trabajo adicional, pero también ayuda a prevenir los siguientes problemas:

- Resultados impredecibles.
- Detección tardía de errores.
- Introducción de herramientas que afecten perjudicialmente al proceso.

Hoy en día, las empresas que realizan esfuerzos para mejorar el proceso de desarrollo de software, se vuelven más eficientes, elevan su nivel de competitividad y aseguran su permanencia en el mercado.

---

<sup>18</sup> Rafael Barzanallana. *Metodologías de desarrollo*. Internet. <<http://www.um.es/docencia/barzana/IAGP/IAGP2-Metodologias-de-desarrollo.html>> Acceso: 18/08/2009.

### 3.1 Metodologías estructuradas

El objetivo de este tipo de metodologías es lograr una definición completa del sistema, estableciendo los datos de entrada y salida para cada una de las funciones que lo componen.

Representan la estructura del sistema de forma jerárquica, van de una visión general hasta llegar a una función específica. Este enfoque jerárquico se puede focalizar en los procesos del sistema, en la estructura de los datos, o en ambos aspectos, dando lugar a metodologías orientadas a procesos, a datos y mixtas.

Se caracterizan por poseer una estructura que consiste en un modelo gráfico jerárquico descendente de los procesos del sistema y de los datos utilizados en dichos procesos. El modelo se compone de:

- *Diagrama de Flujo de Datos (DFD)*: representa los procesos que se llevan a cabo en un sistema y los datos que fluyen a través de él. Los procesos más complejos se descomponen en nuevos diagramas hasta llegar a procesos sencillos.
- *Diccionario de Datos (DD)*: es la definición de todos los datos que se encuentran en el diagrama de flujo de datos.
- *Diagrama entidad-relación*: sirve para modelar la forma en la que los datos están organizados para poder almacenarlos.
- *Diagrama de transición de estados*: modelan el comportamiento del sistema en el tiempo.

Las metodologías estructuradas tratan a los datos como entidades independientes. Estos sistemas resultan difíciles de mantener, ya que si se modifica la estructura de los datos es preciso realizar una revisión completa del sistema para verificar que la modificación se haya propagado correctamente.

Otra desventaja es que la especificación de requerimientos se la realiza en lenguaje común y no en forma estructural, se especifica la funcionalidad que debe tener el sistema y no cómo se deben estructurar los datos.

### **3.2 Metodologías orientadas a objetos**

Este tipo de metodologías se enfoca en modelar un sistema, basándose en los objetos y sus interacciones. Primero se identifican los objetos del sistema para luego especificar su comportamiento.

En la programación orientada a objetos, los sistemas se modelan creando clases, que son un conjunto de datos y funcionalidades. Las clases son definiciones, a partir de las cuales se crean objetos. Los objetos son miembros de una clase determinada y poseen los datos y funcionalidades definidos en esa clase.

Las herramientas que se utilizan para modelar un sistema son:

- *Diagrama de clases:* sirve para describir los componentes del sistema. A diferencia de los diagramas de flujo de datos, muestra relaciones de asociación entre clases, en lugar del flujo de datos entre ellas.
- *Diagrama de casos de uso:* modela un sistema en base al comportamiento que tiene para cada función.
- *Diagrama de transición de estados:* sirve para describir los cambios de estado de los objetos. Es similar a los utilizados en las metodologías estructuradas, pero este se enfoca en los objetos, en lugar de en el sistema.
- *Diagrama de colaboración:* describe la comunicación entre los objetos de un sistema.

Las metodologías orientadas a objetos permiten concebir programas de una manera intuitiva y cercana a la realidad, ya que utilizan lenguaje natural para describir el comportamiento del sistema en vez de hacerlo utilizando funciones.

### **3.3 Metodología para desarrollo Web**

Una metodología de desarrollo Web debe poseer características especiales que le permitan satisfacer los requerimientos propios para proyectos Web, como son: tiempos de entrega cortos, presupuestos definidos y recursos limitados.

Además, debe ser lo suficientemente flexible como para adaptarse a varios tipos de proyectos, que van desde páginas estáticas con propósito informativo, hasta portales Web interactivos.

Para desarrollar el portal inmobiliario “*Vivir en Ecuador*” se utilizará una adaptación de la metodología para desarrollo Web propuesta por la empresa Greynium y que elabora portales Web para clientes de todo el mundo. La metodología original se encuentra en el enlace: <http://www.greynium.com/services/web-development-methodology.html>

Esta metodología permite trabajar con una amplia gama de proyectos Web, involucra al cliente en el proceso de desarrollo y se divide en fases para un seguimiento eficiente del proyecto.

Las cuatro fases que componen la metodología son:

*Fase de Preparación:* Se determina en qué consistirá el proyecto a desarrollar, cuáles serán sus características y los recursos que demandará. Se analizan los requerimientos del cliente para ofrecer una solución óptima y se definen un plazo de entrega y un presupuesto.

Las actividades que se realizan en esta fase son:

- Entrevista con el cliente para acordar especificaciones del proyecto.
- Analizar el alcance del proyecto.
- Elaborar el cronograma de trabajo y el presupuesto.

*Fase de Diseño:* Se define la parte gráfica del proyecto, es importante involucrar al cliente ya que se trata de la imagen este va a proyectar. Como resultado se obtiene una plantilla de diseño que se utilizará para elaborar las páginas que componen el proyecto.

Consta de las siguientes actividades:

- Entrevista con el cliente para definir el diseño.
- Presentar propuestas de diseño.
- Corregir o modificar el diseño.

*Fase de Desarrollo:* Se incorpora la programación al proyecto, se optimizan los elementos gráficos, se crean las bases de datos y las conexiones a estas. Esta fase constituye la puesta en marcha del portal Web.

Las actividades que se ejecutan en esta fase son:

- Elaborar la página de inicio (*homepage*).
- Iniciar la programación del sitio Web.
- Solicitar los contenidos para las páginas del proyecto.
- Publicar el avance del proyecto.
- Incorporar correcciones o modificaciones.

*Fase de Finalización:* Se realiza una revisión final de todo el proyecto y se lo publica en un servidor. Se requiere de la aprobación final del cliente para dar por concluido el proyecto.

Involucra las siguientes actividades:

- Revisión final de contenidos.
- Cargar el proyecto en el servidor Web.
- Efectuar correcciones.
- Aprobación final del cliente.

## **CAPÍTULO IV**

### **ELABORACIÓN DEL PORTAL INMOBILIARIO**

#### **“VIVIR EN ECUADOR”**

La constructora CAPESSO desea realizar un portal Web que además de promocionar sus proyectos de vivienda, proporcione una plataforma para atraer a potenciales clientes y ofrezca a los usuarios registrados la posibilidad de publicar bienes inmuebles.

A continuación se desarrollará la metodología propuesta en el capítulo anterior, que es una adaptación de la metodología utilizada por la empresa Greynium, intentando dar lineamientos para que pueda adaptarse a nuevos proyectos.

La ventaja de esta metodología es que se la puede utilizar en una amplia gama de proyectos, desde páginas Web estáticas hasta gestores de contenido y portales Web. Dependiendo del alcance del proyecto, la metodología puede ser aplicada por una sola persona o un grupo de trabajo.

#### **4.1 Fase de preparación**

El objetivo de esta fase es realizar un acercamiento con el cliente para determinar las especificaciones del proyecto en términos de alcance, tiempo y recursos. Esto proporcionará una idea de la magnitud y la complejidad del proyecto. Es importante brindarle asesoría al cliente, pues no siempre tiene una idea clara de lo que necesita ni de lo que obtendrá, por esta razón se debe explicar con claridad las distintas alternativas que se pueden emplear para cumplir con sus requerimientos.

##### **4.1.1 Entrevista con el cliente para acordar especificaciones del proyecto**

El éxito del proyecto dependerá en gran parte de esta entrevista. Es preciso dar una buena impresión al cliente, así que confirmar reuniones y ser puntuales son aspectos a tener en cuenta.

El objetivo es buscar el equilibrio entre lo que se puede ofrecer y lo que el cliente realmente necesita. Hay que tener en cuenta que no siempre lo más complicado o costoso

satisface plenamente sus requerimientos. En muchas ocasiones, podemos ofrecerle una opción que resulta igual de efectiva pero que demanda menos tiempo y recursos.

Dependiendo de los requerimientos del cliente, se podrá identificar el tipo de sitio Web a elaborar, que pueden ser una página estática, gestores de contenido o portales Web. Además, se podrá conocer qué tipo de información se publicará en la página y también, la estructura general del proyecto.

En el caso de la constructora CAPESSO, deseaba contar con un espacio en *Internet* que le permitiera promocionar sus proyectos de vivienda, pero que además brindara algún servicio a quienes lo visiten. Esto resultaba algo complicado, pues no se trata de una constructora posicionada en el mercado, a la que ubicaran directamente a través de una página Web. Por lo que resultó muy atractiva la idea de crear un portal donde no sólo pudieran ofertar sus proyectos inmobiliarios sino que además pudieran proporcionar un punto de encuentro para personas vinculadas al mercado de los bienes raíces.

El cliente proporcionó dos guías muy claras: el portal debía ser en extremo fácil de usar y visualmente atractivo. No deseaban que tuviera formularios extensos ni complicados de llenar. El reto era diseñar formularios que contengan la información necesaria para crear una base de datos robusta y a la vez trabajar con la menor cantidad de información para que los usuarios no se sientan incómodos al momento de llenarlos. El cliente proporcionó el logotipo que se debía utilizar en el concepto gráfico. La ventaja es que, al ser un proyecto nuevo, se tiene la libertad de elaborar diseños sin restricciones. Es distinto cuando el cliente ya posee una imagen corporativa y el diseño debe adaptarse a ella.

#### **4.1.2 Analizar el alcance del proyecto**

En este punto se define exactamente en qué consistirá el trabajo que se va a realizar, es importante especificar claramente los productos o resultados que se van a entregar y la forma en la que se lo hará. A partir de estas especificaciones se elaborará un cronograma de trabajo y un presupuesto para el proyecto.

Para evitar posibles problemas o en todo caso resolverlos de la mejor manera, es recomendable firmar un contrato de prestación de servicios, en el cuál se detalla: en qué

consiste el proyecto, su costo, el plazo de entrega y las condiciones adicionales, como cláusulas de confidencialidad y derechos de autor. Así, en caso de que exista alguna inconformidad se puede llegar a un acuerdo entre ambas partes.

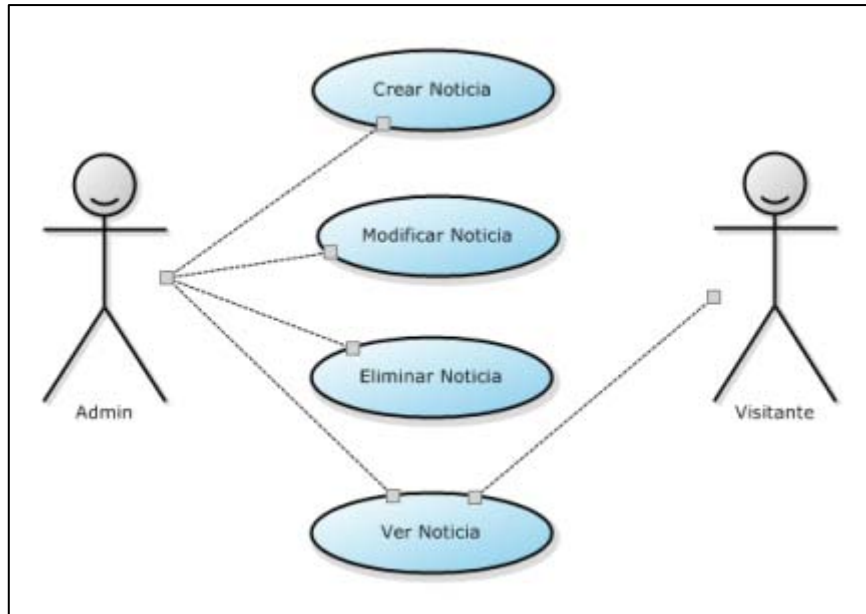
La importancia de especificar con el mayor nivel de detalle las tareas que involucrará el proyecto, se debe a que en el mundo profesional, si omitimos algún detalle que signifique trabajo adicional, deberá ser asumido por el contratista. Esto significa pérdida de tiempo y de dinero.

Basados en la entrevista con el cliente, se elabora un cuadro con los componentes del portal inmobiliario:

<b>Descripción</b>	<b>Componente</b>
Diseño	Diseño gráfico para todo el sitio Web
Páginas Web	<ul style="list-style-type: none"><li>- Inicio</li><li>- Quienes somos</li><li>- Noticias</li><li>- Scalini</li></ul>
Páginas con formularios	<ul style="list-style-type: none"><li>- Regístrese</li><li>- Buscar</li><li>- Contáctenos</li></ul>
Gestor de contenidos	<ul style="list-style-type: none"><li>- Noticias</li></ul>
Módulos dinámicos	<ul style="list-style-type: none"><li>- Usuarios</li><li>- Propiedades</li></ul>
Elementos interactivos	Banner con menú
Hosting y dominio	<ul style="list-style-type: none"><li>- Dominio por un año: <i>www.vivirenecuador.com.ec</i></li><li>- <i>Hosting</i> en servidor <i>Linux</i> por un año: 800 MB</li></ul>

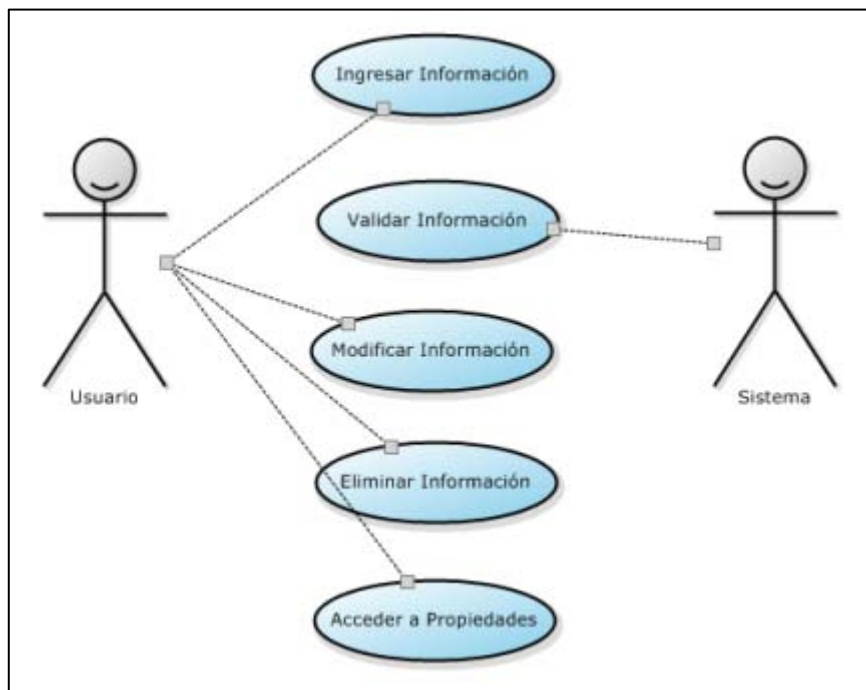
Para definir el alcance del gestor de contenidos y de los módulos dinámicos se utilizarán diagramas de casos de uso:

### Gestor de contenidos para Noticias



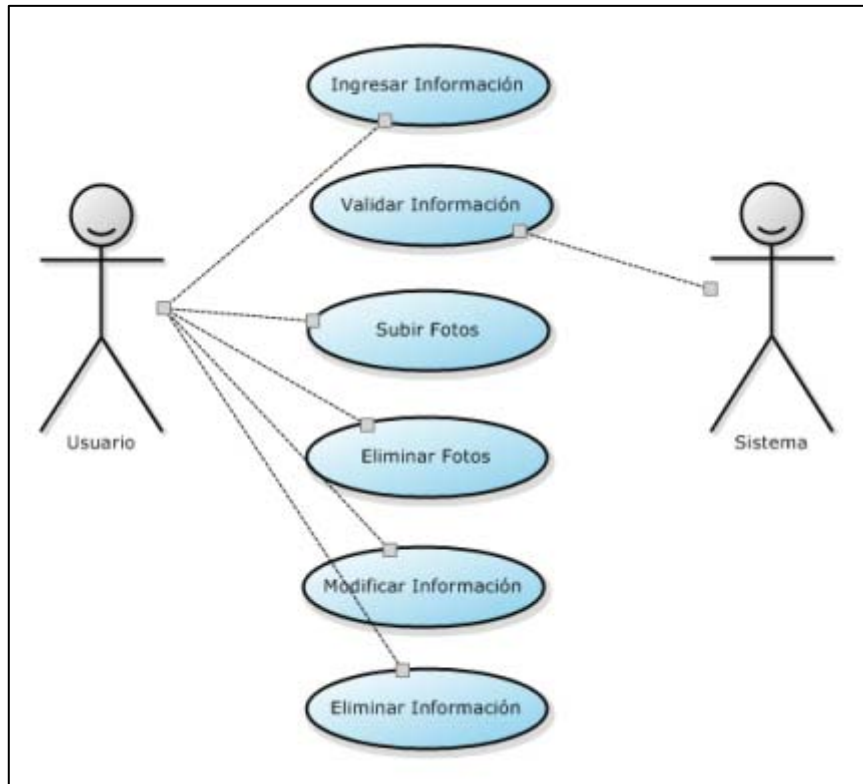
**Figura 4.1. Casos de uso para el gestor de contenidos**  
Fuente: Elaboración propia.

### Módulo de Usuarios



**Figura 4.2. Casos de uso para el módulo de usuarios**  
Fuente: Elaboración propia.

### Módulo de Propiedades



**Figura 4.3. Casos de uso para el módulo de propiedades**

Fuente: Elaboración propia.

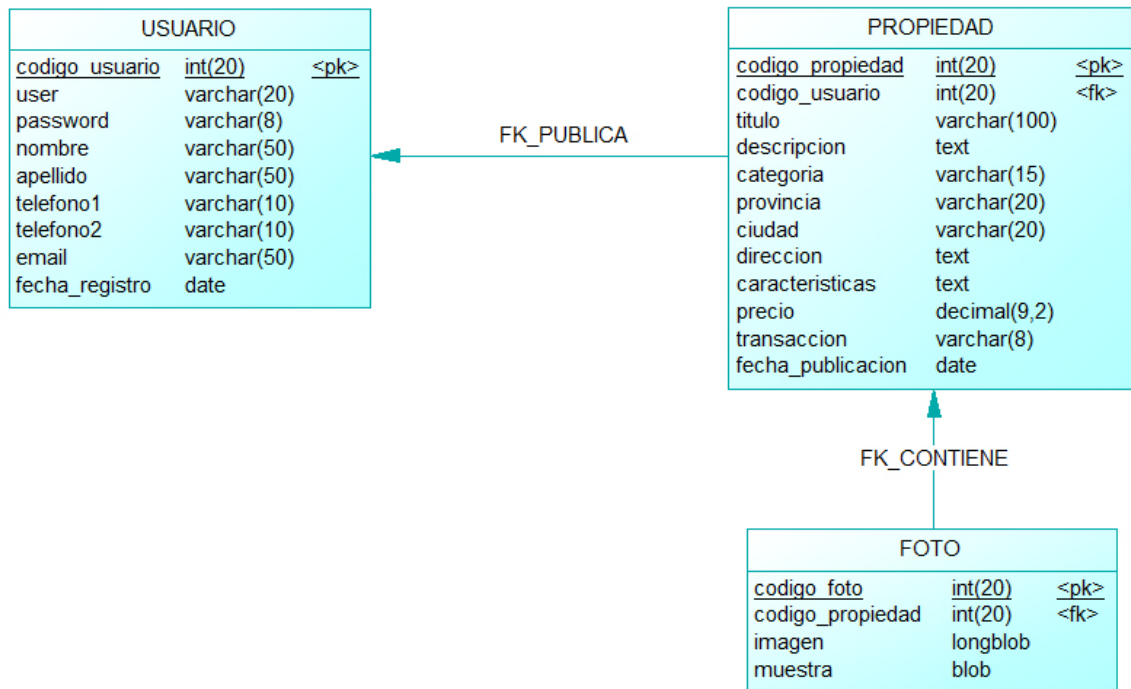
Los visitantes podrán registrarse llenando un formulario de registro, una vez hecho esto podrán modificar sus datos personales, cambiar su contraseña, eliminar su cuenta y acceder al módulo de propiedades. Si un usuario olvida su contraseña, deberá llenar un formulario para confirmar su identidad y enviarle su contraseña a la cuenta de correo que proporcionó al momento de registrarse. La aplicación Web validará la información suministrada por el usuario.

Sólo usuarios registrados podrán publicar propiedades sin límite. Adicionalmente, podrán subir hasta cinco fotografías de su bien inmueble.

El usuario administrador es el único que puede publicar noticias en el portal, en la página de inicio aparecerán las dos noticias más recientes y habrá una página de noticias donde se mostrarán todas las noticias.

Para definir la estructura de la base de datos se utilizarán diagramas entidad-relación de los modelos físicos, que indican el tipo de dato que se usó para el sistema de gestión de bases de datos *MySQL* en su versión 5.1.30.

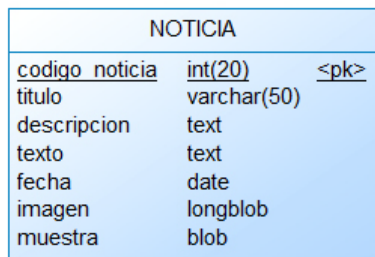
Diagrama entidad-relación para la base de datos:



**Figura 4.4. Diagrama entidad-relación de la base de datos del portal Web**

Fuente: Elaboración propia.

Diagrama entidad-relación para la tabla de noticias:



**Figura 4.5. Diagrama entidad-relación de la tabla noticias**

Fuente: Elaboración propia.

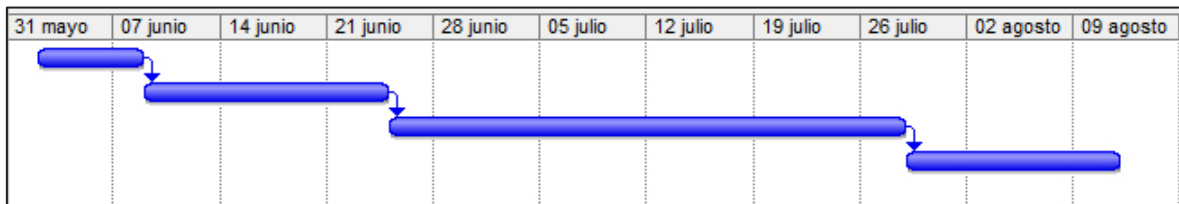
### 4.1.3 Elaborar el cronograma de trabajo y el presupuesto

Una vez definidos los aspectos técnicos del proyecto, se especifica el cronograma de trabajo y se elabora un presupuesto.

Todas las personas están expuestas a sufrir un percance que retrase el desarrollo y la entrega del proyecto, por esta razón, además de tomar precauciones para minimizar el impacto que algún evento fortuito pueda generar sobre el proyecto, es recomendable contar con margen adicional de tiempo, que permitirá solventar cualquier imprevisto.

Cronograma de trabajo para el Portal Inmobiliario “Vivir en Ecuador”

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Fase de preparación	5 días	mar 02/06/09	lun 08/06/09	
2	Fase de diseño	12 días	mar 09/06/09	mié 24/06/09	1
3	Fase de desarrollo	24 días	jue 25/06/09	mar 28/07/09	2
4	Fase de finalización	10 días	mié 29/07/09	mar 11/08/09	3



**Figura 4.6. Diagrama de Gantt para el cronograma de elaboración del portal Web**  
 Fuente: Elaboración propia.

Presupuesto para el Portal Inmobiliario “Vivir en Ecuador”

Descripción	Componente	Costo
Diseño	Diseño gráfico	100,00
Páginas Web	- Inicio	25,00
	- Quienes Somos	25,00
	- Noticias	0,00
	- Scalini	25,00
Páginas con formularios	- Regístrese	0,00
	- Buscar	0,00
	- Contáctenos	35,00
Gestor de contenidos	- Noticias	150,00
Módulos dinámicos	- Usuarios	250,00
	- Propiedades	350,00
Elementos interactivos	Banner con menú	40,00
Hosting y dominio	- www.vivirenecuador.com.ec	35,00
	- Hosting Linux 800MB	90,00
<b>Total</b>		<b>\$ 1.150,00</b>

**Figura 4.7. Detalle del presupuesto para el portal inmobiliario**  
 Fuente: Elaboración propia.

Las páginas con costo cero están incluidas en los módulos, pero se las incluye para especificar que habrá una página accesible para los visitantes del portal.

En este punto ya se tendrá una idea clara de cuánto trabajo demandará el proyecto y el presupuesto que requerirá.

Una vez que el cliente apruebe las especificaciones, el marco de tiempo y el presupuesto del proyecto. Se podrá elaborar un contrato de prestación de servicios como respaldo para ambas partes, de ser necesario, incluso, se puede requerir un anticipo y continuar con la fase de diseño.

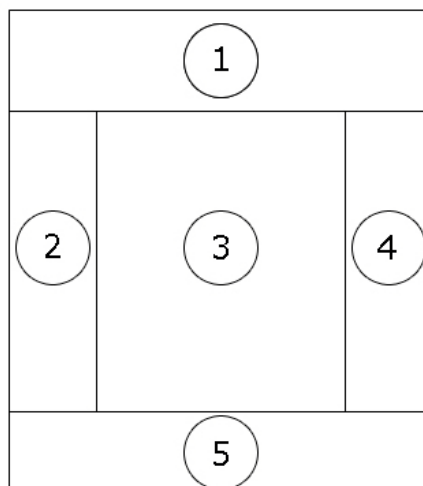
## 4.2 Fase de diseño

El objetivo de esta fase es definir la apariencia y la distribución de las páginas que componen el portal Web. En esta fase es fundamental involucrar al cliente, ya que cambios posteriores en el diseño pueden significar horas de trabajo adicional no remunerado.

### 4.2.1 Entrevista con el cliente para definir el diseño

Tras mantener una entrevista con el cliente, se pudieron definir algunos elementos para el diseño del portal. Todas las páginas tendrán la misma distribución de contenido, lo que facilitará la programación del portal Web, ya que se utilizará una plantilla de diseño para la creación de cada página. Se sugirió además trabajar sobre un fondo claro.

El cliente presentó una definición muy clara en cuanto a la distribución del contenido:



**Figura 4.8. Diagrama de distribución de contenido**

Fuente: Elaboración propia.

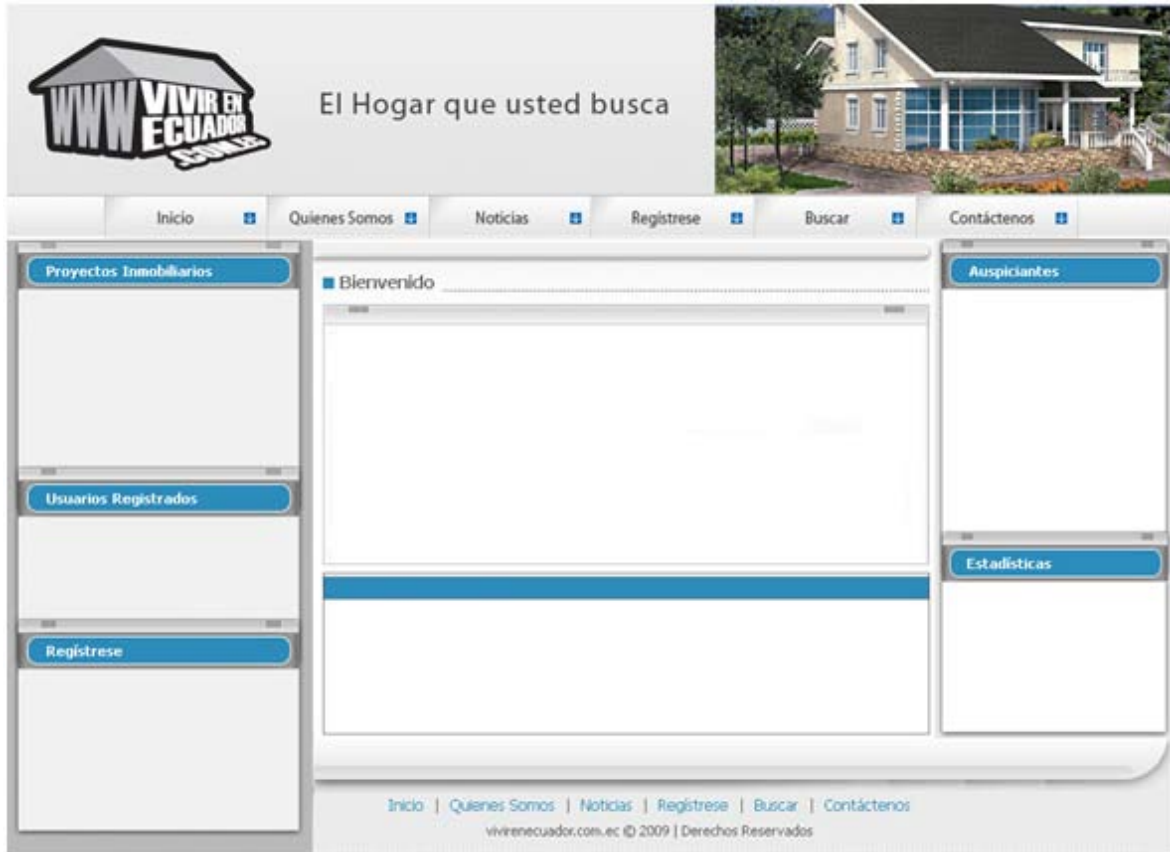
1. *Banner*: incluirá el logotipo, una transición de imágenes y un menú de navegación.
2. *Panel izquierdo*: contendrá una sección para promocionar los proyectos inmobiliarios de la constructora CAPESSO, una sección para que los usuarios registrados se identifiquen y un vínculo a la página de registro de usuarios.
3. *Panel central*: en esta sección se desplegará la información de cada página.
4. *Panel derecho*: incluirá una sección para auspiciantes y una sección para estadísticas, la cual presentará el número de vistas y las vistas recibidas el último mes.
5. *Footer*: incluye un menú básico de navegación.

#### **4.2.2 Presentar propuestas de diseño**

Gracias a que el cliente proporcionó información detallada para el diseño y la distribución del contenido, se elaborará una sola propuesta de diseño. Dependiendo de los requerimientos del cliente se pueden elaborar hasta 3 propuestas de diseño, esto ocurre cuando el cliente no tiene una idea clara de qué tipo de diseño implementará en su sitio Web. Es muy común que se combinen elementos de las diferentes propuestas hasta obtener un resultado satisfactorio.

Las propuestas de diseño son elaboradas en un editor gráfico, el cual provee libertad para la creación de diseños y facilita la realización de cambios. En el caso del portal inmobiliario se utilizó Adobe Photoshop.

La propuesta de diseño que se presentó fue la siguiente:



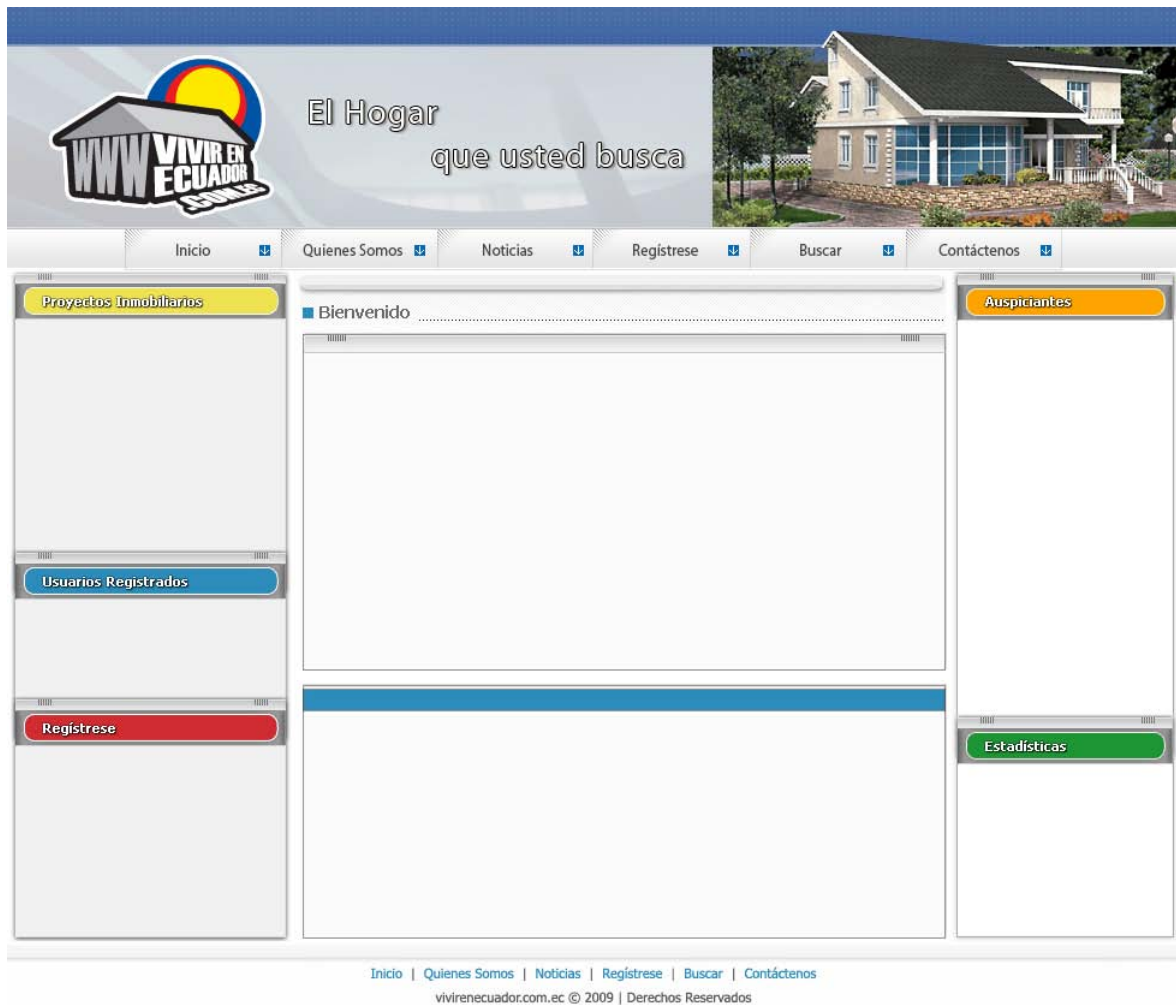
**Figura 4.9. Propuesta inicial de diseño**

Fuente: Elaboración propia.

### 4.2.3 Corregir o modificar el diseño

Como el diseño que se presenta es apenas una propuesta, siempre se realizan correcciones. Se debe procurar orientar al cliente para que este proceso no tome demasiado tiempo. En ocasiones, por tratar de perfeccionar el diseño, se le quita tiempo al resto de fases del proyecto.

Para el objeto de esta disertación, el cliente solicitó que se incluyera más variedad de colores en el diseño, en cuanto a la distribución del contenido no hubo cambios y después de una segunda entrevista con el cliente, éste fue el resultado final:



**Figura 4.10. Propuesta final de diseño**

Fuente: Elaboración propia.

Una vez que se obtiene la aprobación del cliente, se extraen elementos gráficos para generar una plantilla *HTML*, que servirá de modelo para todas las páginas que componen el sitio Web.

### 4.3 Fase de desarrollo

Una vez aprobado el diseño se procede a elaborar una plantilla en *HTML*. Por lo general, el diseño se lo elabora en un editor gráfico, que permite extraer y optimizar los elementos gráficos que componen el diseño, como si se tratara de piezas de un rompecabezas. De esta forma, cuando una página Web se carga en el computador de un visitante, el tiempo que toma visualizar elementos individuales es menor que el tiempo que tomaría cargar el diseño en su totalidad.

#### **4.3.1 Elaborar la página de inicio (*homepage*)**

La página de inicio servirá como modelo para la elaboración del resto de páginas Web que componen el portal, sin importar si éstas son estáticas o dinámicas. Se podrá apreciar una de las ventajas de trabajar con el lenguaje de programación *PHP*, ya que permite incluir sentencias directamente en el código *HTML*.

Para trasladar el diseño a una página *HTML*, se crean tablas que incluirán los elementos gráficos que previamente se extrajeron y optimizaron. Estos se incluyen como fondos de las tablas, de esta manera podemos agregar texto sobre ellos sin ningún problema.

En este punto es conveniente crear una hoja de estilo, la cual contiene los diferentes tipos de texto que se utilizarán en la página. Además de constituir un estándar para la elaboración de páginas Web, es de gran ayuda ya que si realizamos un cambio, como por ejemplo en el color de letra, este se propaga automáticamente a todo el sitio Web.

Si el cliente no ha proporcionado los textos e imágenes para el sitio Web, se utilizará texto de prueba, generado por computadora, e imágenes de muestra. El objetivo es tener un prototipo lo más cercano posible a la realidad y una idea clara de cómo estarán distribuidos el texto y las imágenes en la página Web.

Se deben realizar pruebas locales para verificar que la página de inicio se visualice de forma adecuada, ya que servirá como modelo para generar las demás páginas del proyecto.

#### **4.3.2 Iniciar la programación del sitio Web**

La programación de un sitio Web, dependiendo del tamaño y la complejidad, puede estar a cargo de una persona o de un equipo. Si se conservan estándares en cuanto a diseño y programación, la integración final no será una tarea que demande mayor trabajo.

Para la elaboración del portal inmobiliario “*Vivir en Ecuador*” se utilizó el editor *HTML Adobe Dreamweaver*, que además de proveer un entorno amigable para la elaboración de páginas Web, incorpora herramientas útiles como controles para validación, lo cual agiliza la fase de programación.

Hay que tomar en cuenta que en este punto no deberían existir dudas sobre el diseño o a la estructura del portal Web, ya que un cambio, por más pequeño que sea, puede generar trabajo adicional y retrasos en el plazo de entrega.

Este trabajo no se enfoca en la parte de programación, sin embargo, la bibliografía incluye guías paso a paso de cómo programar una aplicación Web con bases de datos. Además, la ventaja de trabajar con herramientas de software libre es que existe gran cantidad de código fuente disponible en *Internet*, desde cómo programar un formulario de contacto, hasta cómo generar contenido dinámico desde una base de datos.

En los anexos 1 y 2, respectivamente, se incluyen el código fuente de la página de inicio y las instrucciones para generar la base de datos.

A menudo, la parte de programación resulta ser una tarea complicada y tediosa, pero a medida que se adquiere experiencia y gracias al uso de estándares, se puede reutilizar gran parte del código programado anteriormente.

La programación que involucra al usuario debe estar enfocada, sobre todo, en la facilidad de uso y en evitar que el usuario cometa errores. Para el programador, muchos procedimientos son evidentes, pero tratándose de usuarios no se debe asumir su comportamiento, se lo debe inducir con la ayuda de gráficos y restricciones.

### **4.3.3 Solicitar los contenidos para las páginas del proyecto**

En caso de que el cliente aún no haya proporcionado el contenido, tanto texto como imágenes, es el momento de solicitárselos. Hasta ahora hemos trabajado con contenido de prueba, pero antes de subir el avance del proyecto al servidor Web, debemos contar con la información, de ser posible, definitiva.

A menudo, esta tarea genera retrasos en el desarrollo del proyecto, pues el cliente no siempre se da el tiempo de preparar la información. Cuando se trabaja con empresas es recomendable requerir que se asigne una persona responsable de proporcionar la información, de lo contrario, el proyecto puede dilatarse por semanas e incluso meses hasta contar con el contenido para las páginas Web.

Una vez que el cliente proporciona la información, se reemplaza el contenido de prueba por el suministrado. Gracias a que se elaboró un prototipo y a la utilización de hojas de estilo, esta tarea se vuelve sencilla.

#### **4.3.4 Publicar el avance del proyecto**

Utilizaremos un servidor Web destinado para realizar pruebas, ya que no se puede poner en riesgo el servidor del cliente, si es que cuenta con uno, y si no, contratará el servicio de *hosting*<sup>19</sup> en una etapa posterior.

Para la publicación del portal Web en *Internet* utilizaremos un cliente *FTP*, el cual permite transferir archivos a un servidor. De esta manera podemos copiar todos los archivos necesarios para el funcionamiento de un sitio Web en pocos minutos.

Una vez que el proyecto esté alojado en un servidor Web se proporciona al cliente una dirección Web para que pueda observar el avance del portal. Si bien es cierto el proyecto no será 100% funcional, le dará al cliente una versión cercana al resultado final.

El realizar pruebas en un ambiente Web es de gran utilidad ya que, además de corregir errores, permite optimizar el sitio Web. El resultado final no sólo deberá verse bien, sino que también deberá poseer tiempos de respuesta adecuados.

#### **4.3.5 Incorporar correcciones o modificaciones**

En esta etapa, el cliente propone cambios en cuanto a apariencia, se realizan ajustes en el tipo y el tamaño de la letra, se ajusta la distribución del texto o se sustituyen imágenes. Son tareas que no demandan mucho esfuerzo y generan un alto grado de satisfacción en el cliente.

Es recomendable limitar el número de cambios en este punto, ya que si el cliente tiene la posibilidad de realizar cambios ilimitados, los hará. Esto, además de retrasar el tiempo de

---

<sup>19</sup> *Hosting*: servicio que consiste en ceder espacio en un servidor para subir (alojar) un sitio web para que pueda ser accedido en todo momento. Tomado de EXCEPTION *Diccionario Informático* <[http:// www.alegsa.com.ar/Dic/hosting.php](http://www.alegsa.com.ar/Dic/hosting.php)> Acceso: 12/08/2009.

entrega del proyecto, no permite que se sigan realizando pruebas en las funcionalidades del sitio Web.

#### **4.4 Fase de finalización**

En esta fase, se ultiman los detalles del proyecto para su versión definitiva. Se prueban todas las funcionalidades del portal, se corrigen detalles que pueden haberse pasado por alto y se publica en *Internet* la versión final del sitio Web.

##### **4.4.1 Revisión final de contenidos**

Tras realizar las correcciones propuestas por el cliente, se prueban una por una las especificaciones del proyecto. Es recomendable, contar con usuarios que ayuden a verificar que el portal funcione correctamente. Se realizan pruebas en vivo, se le pide a los usuarios que se registren y publiquen una o varias propiedades. En caso de existir alguna anomalía, se la debe documentar para darle seguimiento y corregirla de inmediato.

El cliente recibe capacitación para operar el gestor de contenidos, se proporcionan nombres de usuario y contraseñas, con las que accederá al módulo de noticias. Además, verifica el texto y las imágenes que serán publicados en la versión final.

##### **4.4.2 Cargar el proyecto en el servidor Web**

Una vez realizadas las correcciones para los problemas identificados en la revisión final, el sitio Web está listo para ser cargado en el servidor donde se alojará de forma definitiva. Este proceso incluye la transferencia de todos los archivos necesarios para el funcionamiento del portal y la exportación de las bases de datos hacia el nuevo servidor.

Dependiendo del servidor donde se aloje el proyecto, pueden surgir algunos problemas de compatibilidad, principalmente relacionados con la base de datos. Estos problemas ocurren debido a que existe diferencia entre la versión del sistema de gestión de bases de datos utilizado en las pruebas y el instalado en el servidor definitivo.

#### **4.4.3 Efectuar correcciones**

Se presentaron errores de compatibilidad en la base de datos, para solventarlos se modificó el tipo de dato que ocasionaba conflicto. Se actualizó el diagrama entidad-relación y el código para la generación de la base de datos.

Este tipo de defectos no es común y ocurren en tipos de datos para funciones específicas. En el caso del portal inmobiliario, se presentaron errores de compatibilidad en el tipo de dato asignado a imágenes, ya que la quinta versión de *MySQL* no soporta el tipo de dato *medium blob*, que si es soportado por su cuarta versión. Se solucionó el error cambiando este tipo de dato a *blob*.

#### **4.4.4 Aprobación final del cliente**

En este punto, el portal Web está listo para su lanzamiento oficial. Se realiza una inspección final con el cliente y se solicita su aprobación definitiva. Es recomendable suscribir un documento que respalde la entrega del proyecto y la conformidad de ambas partes.

De esta forma, el proceso de elaboración del portal Web llega a su fin.

## CAPÍTULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 Conclusiones

- Tener presencia en *Internet* ha dejado de ser un lujo para convertirse en una necesidad y, a menudo, es la primera impresión que proyecta una empresa.
- Un portal Web constituye un punto de partida para el usuario, donde puede encontrar contenido organizado y donde se concentran productos y servicios.
- Los portales Web permiten al usuario agregar y modificar información a través de un gestor de contenido y proveen un alto nivel de interactividad.
- El uso de la *Web 2.0* está orientado a las redes sociales. Los sitios Web que implementan esta tecnología le permiten al usuario personalizar el contenido que se despliega en una página Web.
- La ventaja del software libre es que un programador cuenta con el respaldo de una comunidad, la cual provee elementos necesarios para que pueda desarrollar nuevas y mejores soluciones, para luego compartirlas.
- Una herramienta de software libre tiene un desarrollo más acelerado que el software comercial, ya que continuamente, miles de usuarios contribuyen con mejoras.
- La importancia de una metodología radica en que provee un entorno disciplinado y estructurado, que facilita el trabajo y la creación de productos con alta calidad.
- No existe una zonificación clara y precisa para el Distrito Metropolitano de Quito, en muchas ocasiones el barrio resulta ser una referencia inadecuada, pues se necesita un mayor nivel de detalle.

## 5.2 Recomendaciones

- El éxito de un portal Web depende en gran parte del número de vistas que recibe, por lo que resulta importante complementar el desarrollo de un proyecto Web, con técnicas de posicionamiento o con la ayuda de redes sociales, que ayuden a incrementar el número de visitantes.
- Se podría incorporar un sistema de información geográfica (SIG), para proporcionar al usuario una herramienta que le ayude a ubicar con exactitud un bien inmobiliario dentro de un mapa y le permita contar con referencias geográficas.
- Además de PHP, existen otros lenguajes de programación orientados a desarrollo Web. Resultaría interesante migrar el portal desarrollado en PHP hacia nuevos lenguajes de programación como Ruby.

## BIBLIOGRAFÍA

### *Libros*

- Barranco de Areba, Jesús. *Metodología del análisis estructurado de sistemas*. Madrid, Universidad Pontificia Comillas de Madrid, 2da edición, 2001.
- Converse, Tim. *PHP5 and MySQL Bible*. Indianapolis, Wiley Publishing, Inc., 2004.
- Davis, Michele. *Learning PHP and MySQL*. Sebastopol, O'Reilly Media, Inc., 2006.
- Gómez de Silva Garza, Andrés. et al. *Introducción a la computación*. México D.F., Cengage Learning, 2008.
- Greenspan, Jay. *MySQL/PHP Database Applications*. Foster City, IDG Books Worldwide, Inc., 2001.
- Meloni, Julie. *Sams Teach Yourself PHP, MySQL and Apache in 24 Hours*. Indianapolis, Sams Publishing, 2003.
- Valade, Janet. *PHP & MySQL For Dummies*. Indianapolis, Wiley Publishing, Inc., 2da edición, 2004.
- Valade, Janet. *PHP 5 For Dummies*. Indianapolis, Wiley Publishing, Inc., 2004.
- Vaswani, Vikram. *How to Do Everything with PHP & MySQL*. New York, McGraw-Hill, 2005.
- Welling, Luke. *PHP and MySQL Web Development*. Indianapolis, Sams Publishing, 3ra edición, 2005.
- Williams, Hugh. *Web Database Applications with PHP & MySQL*. Sebastopol, O'Reilly Media, Inc., 2da edición, 2004.

- Yank, Kevin. *Build Your Own Database-Driven Website Using PHP & MySQL*. Richmond, SitePoint Pty. Ltd., 2da edición, 2003.
- Zabir, Omar. *Building a Web 2.0 Portal with ASP.NET 3.5*. Sebastopol, O'Reilly Media, Inc., 2008.

#### *Internet*

- Barzanallana, Rafael. *Metodologías de desarrollo*. Internet. <http://www.um.es/docencia/barzana/IAGP/IAGP2-Metodologias-de-desarrollo.html>. Acceso: 18/08/2009.
- CERN. *The Website of the world's first-ever Web server*. Internet. <http://info.cern.ch/> Acceso: 08/08/2009.
- Daccach, José. *Tendencias para el 2009*. Internet. <http://www.deltaasesores.com/articulos/tecnologia/1420-tendencias-para-el-2009>. Acceso: 18/08/2009.
- DesarrolloWeb. *Qué es PHP*. Internet. <http://www.desarrolloweb.com/articulos/392.php>. Acceso: 24/02/2009.
- De Silva, Dharshin. *Paging Results With PHP and MYSQL*. Internet. <http://dharshin.freehostia.com/node/8>. Acceso: 24/02/2009.
- EXCEPTION. *Diccionario Informático*. Internet. <http://www.alegsa.com.ar/Dic/javascript.php>. Acceso: 22/08/2009.
- Giordana, Juan Francisco. *PHP Login Script... the right way*. Internet. <http://www.juangiordana.com.ar/blog/2006/11/28/php-login-script/>. Acceso: 24/02/2009.
- Gracia, Joaquin. *CMM - CMMI*. Internet. <http://www.ingenierossoftware.com/calidad/cmm-cmmi.php>. Acceso: 18/08/2009.

- Greynium. *Web Development Methodology*. Internet. <http://www.greynium.com/services/web-development-methodology.html>. Acceso: 24/02/2009.
- Grupo Gesfor. *La nueva generación de portales: de la información a la colaboración*. Internet. <http://www.grupogesfor.com/htm/02/CasosExito/ECMyWeb2.0/Web2.0.pdf>. Acceso: 08/08/2009.
- MacManus, Richard. *The Evolution of Corporate Web Sites*. Internet. [http://www.digital-web.com/articles/the\\_evolution\\_of\\_corporate\\_web\\_sites/](http://www.digital-web.com/articles/the_evolution_of_corporate_web_sites/). Acceso: 08/08/2009.
- Millán, José. *Vocabulario de ordenadores e Internet*. Internet. [http://jamillan.com/v\\_cookie.htm](http://jamillan.com/v_cookie.htm). Acceso: 22/08/2009.
- Miniwatts Marketing Group. *South America Internet Usage and Population Statistics*. Internet. <http://www.internetworldstats.com/south.htm>. Acceso: 24/02/2009.
- Netcraft. *Usage Stats for April 2007*. Internet. <http://www.php.net/usage.php>. Acceso: 15/08/2009.
- Netcraft. *August 2009 Web Server Survey*. Internet. [http://news.netcraft.com/archives/2009/08/1/august\\_2009\\_Web\\_server\\_survey.html](http://news.netcraft.com/archives/2009/08/1/august_2009_Web_server_survey.html). Acceso: 15/08/2009.
- Official Google Blog. *We knew the Web was big...* Internet. <http://googleblog.blogspot.com/2008/07/we-knew-Web-was-big.html>. Acceso: 08/08/2009.
- Software Engineering Institute. *Capability Maturity Model Integration*. Internet. <http://www.sei.cmu.edu/cmmi/>. Acceso: 18/08/2009.
- Stallman, Richard. *Por qué «software libre» es mejor que «código abierto»*. Internet. <http://www.gnu.org/philosophy/free-software-for-freedom.es.html>. Acceso: 12/08/2009.

- The GNU Project. *¿Qué es Copyleft?*. Internet. <http://www.gnu.org/copyleft/copyleft.es.html>. Acceso: 20/08/2009.
  
- U.S. Department of Health and Human Services. *Research-Web Based Design & Usability Guidelines*. Internet. <http://www.usability.gov/pdfs/guidelines.html>. Acceso: 24/02/2009.
  
- Well House Consultants Ltd. *Using PHP and MySQL to Provide an Image Library*. Internet. <http://www.wellho.net/solutions/php-example-php-form-image-upload-store-in-mysql-database-retrieve.html>. Acceso: 24/02/2009.

## ANEXOS

### Anexo 1:

Código fuente de la página de inicio

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Vivir en Ecuador - Inicio</title>
<style type="text/css">
<!--
body {
    background-image: url(images/bg.jpg);
    background-repeat: repeat;
}
-->
</style>
<link href="style.css" rel="stylesheet" type="text/css" />
<script src="Scripts/AC_RunActiveContent.js" type="text/javascript"></script>
</head>

<body>
<table width="976px" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="933"></td>
</tr>
<tr>
<td bgcolor="#FBFBFB"><table width="976" border="0" cellspacing="0"
cellpadding="0">
<tr>
<td width="239" valign="top"><table width="239" border="0" cellpadding="0"
cellspacing="0" background="images/lbg.jpg">
<tr>
<td colspan="3"></td>
</tr>
<tr>
<td width="13">&nbsp;</td>
<td width="213" height="240" valign="middle"></td>
<td width="13">&nbsp;</td>
</tr>
<tr>
<td colspan="3"></td>
```

```
</tr>
<tr>
<td>&nbsp;&nbsp;&nbsp;</td>
<td>
<table width="213" border="0" cellspacing="0" cellpadding="0">
<form action="login.php" method="post" name="login" id="login">
<tr>
<td width="30%" height="32" class="texto">Usuario</td>
<td width="59%"><input name="username" type="text" class="texto"
style="width:112px;" maxlength="30" /></td>
<td width="11%">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td height="32" class="texto">Contraseña</td>
<td><input name="password" type="password" class="texto"
style="width:112px;" maxlength="12" /></td>
<td valign="middle">
<div align="left">
<input name="image22" type="image" src="images/button.gif" />
</div></td>
</tr>
</form>
</table>

</td>
<td>&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td colspan="3"></td>
</tr>
<tr>
<td>&nbsp;&nbsp;&nbsp;</td>
<td>
<center>
<a href="registro.php"></a>
</center>
</td>
<td>&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td colspan="3"></td>
</tr>
</table></td>
<td valign="top" width="547"><table width="547" border="0" cellpadding="0"
cellspacing="0" background="images/cbg.jpg">
<tr>
<td colspan="3"></td>
```

```
</tr>
<tr>
  <td width="13">&nbsp;</td>
  <td width="521">
    <iframe name="data" src="ver_noticias.php" width="521px"
height="250px" marginwidth="0" marginheight="0" hspace="0" vspace="0"
frameborder="0"></iframe>
  </td>
  <td width="13">&nbsp;</td>
</tr>
<tr>
  <td colspan="3"></td>
</tr>
<tr>
  <td colspan="3"></td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td valign="top">
    <form name="viajeannual" action="viajeannual.php" method="post"
enctype="multipart/form-data">
      <table width="521" border="0" cellpadding="0" cellspacing="0" class="texto">
        <tr>
          <td height="10" colspan="5">&nbsp;</td>
        </tr>
        <tr>
          <td width="65" height="30">Categoría</td>
          <td width="175">
            <select name="categoria" class="texto" style="width:175px;">
              <option value="0" selected>Seleccione una Categoría
              <option value="casa">Casa
              <option value="departamento">Departamento
              <option value="oficina">Oficina
              <option value="local">Local Comercial
              <option value="terreno">Terreno
            </select>
          </td>
          <td width="41">&nbsp;</td>
          <td width="65">Ordenar Por</td>
          <td width="165">
            <select name="ordenar" class="texto" style="width:175px;">
              <option value="precio" selected>Precio
              <option value="fecha">Fecha de Publicación
            </select>
          </td>
        </tr>
        <tr>
          <td height="30">Provincia</td>
          <td>
            <select name="provincia" class="texto" style="width:175px;">
              <option value="0" selected>Seleccione una Provincia
```

```
        <option value="Pichincha">Pichincha
    </select>        </td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>
    <select name="por" class="texto" style="width:175px;">
        <option value="asc" selected>Ascendente
        <option value="desc">Descendente
    </select>        </td>
</tr>
<tr>
<td height="30">Ciudad</td>
<td>
    <select name="ciudad" class="texto" style="width:175px;">
        <option value="0" selected>Seleccione una Ciudad
        <option value="Quito">Quito
    </select>        </td>
<td>&nbsp;</td>
<td>Transacción</td>
<td>
    <select name="por" class="texto" style="width:175px;">
        <option value="asc" selected>Ascendente
        <option value="desc">Descendente
    </select>        </td>
</tr>
<tr>
<td height="40" colspan="5" valign="bottom">
    <div align="center">
        <input type='submit' name='buscar' value='  Buscar  ' />
    </div></td>
</tr>
</table>
</form>
</td>
<td>&nbsp;</td>
</tr>
<tr>
<td colspan="3"></td>
</tr>

</table></td>
<td valign="top"><table width="190" border="0" cellpadding="0" cellspacing="0"
background="images/rbg.jpg">
<tr>
<td colspan="3"></td>
</tr>
<tr>
<td width="13">&nbsp;</td>
<td width="164" height="237"><p>&nbsp;</p>
```

```
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p></td>
<td width="13">&nbsp;</td>
</tr>
<tr>
<td colspan="3"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td><p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p></td>
<td>&nbsp;</td>
</tr>
<tr>
<td colspan="3"></td>
</tr>
</table></td>
</tr>

</table></td>
</tr>
<tr>
<td></td>
</tr>
</table>

<map name="Map" id="Map"><area shape="rect" coords="269,17,317,41"
href="index.php" />
<area shape="rect" coords="318,17,422,41" href="quienes.php" />
<area shape="rect" coords="423,17,484,41" href="noticias.php" />
<area shape="rect" coords="485,17,561,41" href="registro.php" />
<area shape="rect" coords="562,17,617,41" href="buscar.php" />
<area shape="rect" coords="618,17,708,41" href="contactenos.php" />
</map></body>
</html>
```

## Anexo 2:

Instrucciones para generar la base de datos

```
-- phpMyAdmin SQL Dump
-- version 2.11.9.5
-- http://www.phpmyadmin.net
--
-- Servidor: localhost
-- Tiempo de generación: 25-09-2009 a las 04:20:04
-- Versión del servidor: 5.1.30
-- Versión de PHP: 5.2.9

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Base de datos: `vivirene_bdd`
--
-----

--
-- Estructura de tabla para la tabla `FOTO`
--

CREATE TABLE IF NOT EXISTS `FOTO` (
  `FOTO_CODIGO` bigint(20) NOT NULL AUTO_INCREMENT,
  `PROPIEDAD_CODIGO` bigint(20) DEFAULT NULL,
  `FOTO_IMAGEN` mediumblob NOT NULL,
  `FOTO_MUESTRA` blob NOT NULL,
  PRIMARY KEY (`FOTO_CODIGO`),
  KEY `RELATIONSHIP_2_FK` (`PROPIEDAD_CODIGO`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-----

--
-- Estructura de tabla para la tabla `PROPIEDAD`
--
```

```
CREATE TABLE IF NOT EXISTS `PROPIEDAD` (  
  `PROPIEDAD_CODIGO` bigint(20) NOT NULL AUTO_INCREMENT,  
  `USUARIO_CODIGO` bigint(20) DEFAULT NULL,  
  `PROPIEDAD_TITULO` varchar(100) NOT NULL DEFAULT "",  
  `PROPIEDAD_DESCRIPCION` text NOT NULL,  
  `PROPIEDAD_CATEGORIA` varchar(15) NOT NULL DEFAULT "",  
  `PROPIEDAD_PROVINCIA` varchar(20) NOT NULL DEFAULT "",  
  `PROPIEDAD_CUIDAD` varchar(20) NOT NULL DEFAULT "",  
  `PROPIEDAD_UBICACION` text NOT NULL,  
  `PROPIEDAD_CARACTERISTICAS` text NOT NULL,  
  `PROPIEDAD_PRECIO` decimal(9,2) NOT NULL DEFAULT '0.00',  
  PRIMARY KEY (`PROPIEDAD_CODIGO`),  
  KEY `RELATIONSHIP_1_FK` (`USUARIO_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

```
-----  
  
--  
-- Estructura de tabla para la tabla `USUARIO`  
--
```

```
CREATE TABLE IF NOT EXISTS `USUARIO` (  
  `USUARIO_CODIGO` bigint(20) NOT NULL AUTO_INCREMENT,  
  `USUARIO_NOMBRE` varchar(50) CHARACTER SET latin1 COLLATE  
latin1_spanish_ci NOT NULL,  
  `USUARIO_APELLIDO` varchar(50) CHARACTER SET utf8 COLLATE  
utf8_spanish_ci NOT NULL,  
  `USUARIO_TELEFONO1` varchar(10) CHARACTER SET latin1 COLLATE  
latin1_spanish_ci NOT NULL,  
  `USUARIO_TELEFONO2` varchar(10) CHARACTER SET latin1 COLLATE  
latin1_spanish_ci DEFAULT NULL,  
  `USUARIO_EMAIL` varchar(50) CHARACTER SET latin1 COLLATE  
latin1_spanish_ci NOT NULL,  
  `USUARIO_USUARIO` varchar(20) CHARACTER SET latin1 COLLATE  
latin1_spanish_ci NOT NULL,  
  `USUARIO_CONTRASENA` varchar(8) CHARACTER SET latin1 COLLATE  
latin1_spanish_ci NOT NULL,  
  `USUARIO_FECHA` date NOT NULL,  
  PRIMARY KEY (`USUARIO_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=22 ;
```